the
AARDVARK JOURNAL
december 1980 vol.1 , no. 5

THIS MONTH - AND A FEW IN THE FUTURE.

THIS MONTH IS RELATIVELY HEAVY ON PROGRAMS. WE ALSO HAVE A COUPLE OF ARTICLES COURTESY OF THE UK USERS GROUP. THERE IS A STRONG USERS GROUP IN ENGLAND WHERE THE C1P IS PARTICULARLY POPULAR. THEY CONTRIBUTED THE "MOVING THE EXTENDED MONITOR" AND "VIDEO MOD" ARTICLES.

WE ARE NOW TAKING SUBSCRIPTIONS FOR A SECOND YEAR. DESPITE MASSIVE INFLATION I'M GOING TO HOLD THE PRICE DOWN TO $9.00 A YEAR. WE DO HAVE TO RAISE THE PRICE FOR FOREIGN SUBSCRIPTIONS (EXCEPT CANADA) AS WE'RE TAKING A BATH ON THOSE. IT COSTS ABOUT 80 CENTS TO MAIL EACH ISSUE TO EUROPE OR AUSTRALIA. SORRY FELLOWS. FOREIGN SUBSCRIPTIONS ARE NOW $14.00

NEXT YEAR THE JOURNAL WILL CONCENTRATE MORE ON MACHINE CODE TECHNIQUES AND THE FIRST ISSUE OF THE NEW YEAR WILL BEGIN A SERIES OF TUTORIALS ON ASSEMBLER PROGRAMMING. WE'LL CONTINUE TO OFFER A LOT OF SOFTWARE AND WILL TRY TO KEEP THE FORMAT ABOUT WHERE IT IS - AT 20 PACKED PAGES PER ISSUE. THAT'S MORE OSI ARTICLES THAN YOU GET IN ANY GENERAL PURPOSE MAGAZINE.

IN THIS ISSUE, THE TUTORIAL IS ON TROUBLE SHOOTING BASIC. THE BEGINNERS CORNER IS ON ON-GOTO PROGRAMMING AND THERE IS SOME REAL GOLD IN THE LETTERS THIS MONTH.

THE REAL TREASURE FOR C1P OWNERS IS A SCHEMATIC FOR A REAL AND PRACTICAL VIDEO MOD. - AND AN OFFER FOR PLANS FOR AN EVEN BETTER ONE.

THE FREE MONITOR ROMS THIS MONTH WENT TO WIDEN (WHO OPTED FOR A C1P SUPERDISK INSTEAD.), AND THE AUTHORS OF THE MODIFICATIONS TO THE C4P CURSOR AND C1P HIGH SPEED TAPE LOADERS. (AS SOON AS WE FIND THE AUTHOR OF THE C4P CURSOR MODIFICATION.) EVERYONE ELSE WHO WAS PUBLISHED (INCLUDING LETTERS TO THE EDITOR.) GOT GIFT CERTIFICATES AVERAGING $20. SEVERAL ARTICLES SUBMITTED LAST MONTH ARE STILL IN THE RUNNING FOR NEXT MONTHS AWARDS. WE WILL AGAIN OFFER FREE MONITOR ROMS FOR THE BEST SUGGESTIONS AND GIFT CERTIFICATES FOR ANY PUBLISHED HINT.

THE BIG NEWS AT AARDVARK IS THAT WE ARE NOW AN OSI DEALER. WE WERE ALREADY INTO HARDWARE PRODUCTION, SOFTWARE AND FIRMWARE AND IT JUST SEEMED LOGICAL TO OFFER THE COMPUTERS TOO. WE ARE GOING TO OFFER DISCOUNT MAIL ORDER PRICES AND SOME NEAT SPECIAL DEALS.

WE ARE STARTING OFF BY OFFERING THE MODEL II SUPERBOARD AT $279 WITH A FREE CURSOR CONTROL PROGRAM TOSSED IN. THIS COULD BE FUN.

** MERRY CHRISTMAS **

IT'S BEEN A GOOD YEAR AROUND HERE. WE'VE ENJOYED BEING A PART OF A BRAND NEW INDUSTRY - AND I HOPE YOUR YEAR WAS AS GOOD.

BASIC TROUBLESHOOTING FOR BASIC

NOW YOU AND I BOTH KNOW THAT BASIC IS SO SIMPLE THAT A CHILD OF FOUR COULD WRITE IT WITH A DAY OR TWO OF TRAINING. THEREFORE, NEITHER ONE OF US HAS EVER HAD AN ERROR, BUT THE POOR FELLOW SITTING NEXT TO YOU, OR THE ONE WHOM YOU PASS THIS COPY ON TO IS NOT QUITE AS PERFECT AS WE ARE. THAT POOR FELLOW TYPED IN A THREE HUNDRED LINE STARTREK PROGRAM OUT OF 'WHAT TO DO AFTER YOU HIT RETURN' AND THE FIRST TIME HE RAN IT HE GOT THE MESSAGE 'FC ERROR IN LINE 915'. HE DIDN'T EVEN KNOW HE HAD A LINE 915. THIS ARTICLE IS FOR HIM, AND THEREFORE YOU AND I CAN GO ON TO MORE INTERESTING THINGS IN THE BACK OF THE JOURNAL.

ALL PUN ASIDE, BASIC IS BASICALLY SIMPLE TO TROUBLESHOOT. THERE ARE, HOWEVER, SOME TRICKS TO UNDERSTANDING THE ERROR CODES AND SOME COMMON THINGS TO LOOK FOR, SO BEFORE WE DISCUSS HOW TO LOOK FOR A PROBLEM LETS DISCUSS WHAT TO LOOK FOR WHEN THE CODE COMES UP.

THE MOST COMMON ERROR IS 'SN' OR SYNTAX ERROR. (YOU FELLOWS WITH THE BASIC CHIPS WILL GET 'S' AND A FUNNY GRAPHICS SYMBOL AS THE ERROR CODE. FOR THIS AND ALL THE OTHER ABBREVIATIONS I'M GOING TO REFER YOU TO PAGE NINE OF OSI'S GRAPHICS MANUAL AND FROM NOW ON WE'LL PRETEND THAT YOU HAVE THAT AND THEREFORE CAN FIGURE OUT WHICH ERROR CODE WE'RE TALKING ABOUT.) IN ANY CASE, SYNTAX ERROR SHOULD LOGICALLY PLACED LAST IN A LIST OF ERROR CODES AND REALLY SHOULD READ 'NONE OF THE ABOVE'. THERE ARE A LOT OF CONDITIONS THAT CAN CAUSE A 'SYNTAX ERROR' BECAUSE BASIC JUMPS TO SYNTAX ERROR WHENEVER IT CAN'T FIGURE OUT WHAT TO DO NEXT, AND CAN'T FIGURE OUT WHY IT CAN'T FIGURE IT OUT. THE TWO MOST COMMON THINGS TO LOOK FOR IN A SYNTAX ERROR PROBLEM ARE MISSPELLINGS OF THE KEY WORDS IN THE LINE SUCH AS 'TEEN' IN PLACE OF 'THEN' AND UNMATCHED SETS OF PARENTHESIS. ANY MATHEMATICAL OR LOGICAL CONSTRUCT MUST (LOGICALLY) HAVE THE SAME NUMBER OF RIGHT PARENTHESIS AS IT DOES LEFT PARENTHESIS. FOR SOME REASON IT IS DIFFICULT FOR ALL COMPUTERISTS, INCLUDING MYSELF, TO KEEP THE COUNT STRAIGHT. THEREFORE, IF YOU HAVE A SYNTAX ERROR IN A LINE WITH A LOT OF PARENTHESIS, COUNT THE NUMBER OF RIGHTS AND THE NUMBER OF LEFTS AND SEE THAT THEY MATCH UP. AFTER MISSPELLINGS OF KEY WORDS AND PARENTHESIS MISMATCH, THE NEXT MOST COMMON CAUSE OF A SYNTAX ERROR IS AN ILLEGAL VARIABLE. FOR INSTANCE, WHEN I WROTE THE HELICOPTER GAME FOR THIS MONTH'S ISSUE, I CARELESSLY ASSIGNED THE TOP OF THE SCREEN TO A VARIABLE CALLED 'TOP'. WHEN BASIC REACHED THAT LINE IT REPORTED A SYNTAX ERROR BECAUSE I HAD THE WORD 'TO' IN THE VARIABLE NAME AND THEREFORE CONFUSED THE BASIC. THE NICEST THING ABOUT A SYNTAX ERROR IS THAT IT IS VIRTUALLY ALWAYS CONTAINED IN THE LINE NAMED BY THE SYSTEM. IF THE SYSTEM SAYS, FOR INSTANCE, 'SN ERROR LINE 900' THEN THE ONLY LINE THAT YOU ARE GOING TO HAVE TO CORRECT, AND THE ONLY POSSIBLE PLACE FOR THE ERROR - AT LEAST FOR THIS ONE INSTANCE - IS IN LINE 900.

'FC' OR FUNCTION CALL ERROR. I REFUSE TO TRY TO GIVE YOU A CLEAR EXPLAINATION OF EXACTLY WHAT A FUNCTION CALL ERROR IS BECAUSE I DON'T BELIEVE I SHOULD TRY TO DO SOMETHING THAT THE PERPERTRATORS OF BASIC OBVIOUSLY DID NOT DO. HOWEVER, A NUMBER OF DIFFERENT CONDITIONS CAN CAUSE A 'FUNCTION CALL ERROR'. GENERALLY, IT IS A MATTER OF TRYING TO CALL AN IMPOSSIBLE SUBSCRIPT OR TO TAKE AN IMPOSSIBLE FUNCTION. FOR INSTANCE, A SUBSCRIPTED VARIABLE MAY ONLY HAVE NON-NEGATIVE SUBSCRIPTS. (THAT MEANS THAT THE LOWEST POSSIBLE SUBSCRIPT IS ZERO.) TRYING TO CALL 'A(-1)' WOULD CAUSE A FC ERROR AS THERE CAN BE NO SUCH VARIABLE. IN A SIMILAR MANNER, TRYING TO TAKE THE ASCII FUNCTION OF A NON-EXISTANT STRING WILL CAUSE A FC ERROR AS THE SYSTEM OBVIOUSLY CAN'T DO THAT. THE BEST HINT I CAN GIVE YOU ABOUT FUNCTION CALL ERRORS IS THAT YOU ARE NORMALLY LOOKING FOR AN IMPOSSIBLE NUMBER OCCURING INSIDE THE SUBSCRIPTS ON A FUNCTION OR SUBSCRIPTED VARIABLE CALL. WHEN YOU GET THE ERROR, PRINT OUT WHATEVER IS INSIDE THE SUBSCRIPTS IN THAT LINE AND MAKE SURE THAT THEY ARE WITHIN A POSSIBLE RANGE.

THE NEXT MOST COMMON ERROR IS PROBABLY THE 'US' OR UNDEFINED STATEMENT ERROR. THAT IS PRETTY EASY TO UNDERSTAND AS IT SIMPLY THAT YOU ASKED THE SYSTEM TO GO TO A NON-EXISTANT LINE NUMBER. WHEN THAT HAPPENS, LOOK IN THE PROGRAM FOR SIMILAR LINE NUMBERS, AS THE MOST COMMON PROBLEMS ARE DROPPING THE LAST DIGIT OFF A LINE NUMBER OR TYPING A LINE NUMBER INTO THE COMPUTER ACCIDENTLY WITH NOTHING AFTER IT AND THEREFORE ERASING THAT LINE. THERE IS, HOWEVER, NOTHING DIFFICULT ABOUT TROUBLESHOOTING A 'US' ERROR.

ANOTHER FAIRLY EASY ONE TO HANDLE IS THE 'TM' OR TYPE MISMATCH ERROR. ALL A 'TM' ERROR MEANS IS THAT YOU HAVE TRIED TO TREAT A VARIABLE AS IF IT WERE A STRING OR A STRING AS IF IT WERE A VARIABLE. THERE ARE TWO COMMON PROBLEMS TO LOOK FOR WHEN YOU COME TO A 'TM' ERROR. THE MOST COMMON THING IS LEAVING THE '$' SIGN OFF A VARIABLE WHICH YOU MEANT TO BE A STRING, AND THE NEXT MOST COMMON ERROR IS LEAVING OFF THE QUOTATION MARK IN THE FRONT OF STRINGS, MAKING BASIC

"THE AARDVARK JOURNAL" IS PUBLISHED SIX TIMES YEARLY BY RODGER OLSEN, AARDVARK TECHNICAL SERVICES, 1690 BOLTON, WALLED LAKE, MI 48088. SUBSCRIPTION RATE $9.00 , OVERSEAS $14.00. COPYRIGHT 1980 BY RODGER OLSEN. BULK RATE POSTAGE (PERMIT #5) PAID AT WALLED LAKE, MI 48088

THINK THAT IT IS A VARIABLE VALUE RATHER THAN A STRING. IT IS SIMPLE ENOUGH, WITH A 'TM' ERROR LOOK FOR A MISSING '$' OR A MISSING QUOTE MARK.

I WOULD GUESS THAT THE NEXT MOST FREQUENT ERROR IS THE 'CN' OR CONTINUE ERROR. AS WE ARE GOING TO DISCUSS LATER, ONE OF THE NICEST THINGS ABOUT BASIC IS THAT YOU CAN STOP THE PROGRAM RIGHT IN THE MIDDLE OF EXECUTION, PRINT OUT SOME VARIABLES, LOOK AT THE PROGRAM, AND THEN CONTINUE RIGHT ON AS THOUGH THE PROGRAM HAD NEVER STOPPED. THERE ARE, HOWEVER, SOME LIMITATIONS ON THAT ABILITY. IF YOU CHANGE ANY ONE OF THE LINES IN THE PROGRAM, THE SYSTEM WILL LOSE ALL OF THE VARIABLES AND WILL BE UNABLE TO CONTINUE. THAT WILL GIVE YOU A 'CN' ERROR. ALSO, ANY SYNTAX ERROR FOLLOWING THE BREAK OR STOP WILL RENDER BASIC UNABLE TO CONTINUE. IN THAT CASE, YOU CAN GO ON TO THE NEXT LINE BY SIMPLY TYPING IN 'GOTO...' AND THE NEXT LINE NUMBER IN THE PROGRAM.

NEXT, THE 'DD' OR DOUBLE DIMENSION ERROR. THE ONLY CONFUSING THING ABOUT THIS TO A NEW COMPUTERIST IS THAT OFTEN OCCURS THE FIRST TIME THAT YOU DIMENSION A VARIABLE. IF YOU PUT IN 'DIMA(10)' AND GET BACK A 'DD' ERROR, WHAT HAS ACTUALLY HAPPENED IN MOST DOUBLE DIMENSION ERRORS IS THAT YOU HAVE USED THAT SUBSCRIPTED VARIABLE NAME SOMEWHERE PREVIOUSLY IN THE PROGRAM AND BASIC HAS ASSUMED A DIMENSION ASSIGNMENT FOR IT. AS WE DISCUSSED IN LAST MONTH'S ARTICLE ON SUBSCRIPTED ARRAYS, IF BASIC SEES A SUBSCRIPTED VARIABLE AND HAS NOT BEEN TOLD HOW BIG AN ARRAY TO PUT UP FOR IT, IT WILL ASSUME AN ARRAY OF SIZE TEN AND DIMENSION IT AUTOMATICALLY. IF IT THEN COMES TO A DIMENSION STATEMENT WITH THAT VARIABLE IN IT IT COMES UP WITH A 'DD' ERROR AS IT ALREADY HAS AN ASSUMED DIMENSION. WHAT HAS ACTUALLY HAPPENED IN MOST CASES IS THAT WE FINISH THE PROGRAM, GET IT ALMOST RUNNING, AND THEN REALIZE THAT WE NEED TO DO SOMETHING IN LINE 10, AND ACCIDENTLY MODIFY THE PROGRAM SO THAT WE PUT IN A(X) IN THE PROGRAM BEFORE THE DIMENSION STATEMENTS.

'NF', NEXT WITHOUT FOR, OFTEN CAUSES CONFUSION BECAUSE IT CAN OCCUR WHEN THERE IS NOT A DARNED THING LOGICALLY WRONG WITH THE BASIC PROGRAM. AS THE NAME IMPLIES, THE 'NF' ERROR MEANS THAT THE COMPUTER HAS FOUND A NEXT STATEMENT AND DOESN'T KNOW NEXT WHAT. THE ERROR OCCURS IN TWO COMMON WAYS ONE OF WHICH IS THE COMPUTERIST'S FAULT AND ONE THE MACHINE'S FAULT. YOU NORMALLY FIND A NEXT WITHOUT FOR ERROR OCCURING WHEN YOU HAVE SEVERAL NESTED FOR-NEXT LOOPS. THE MOST COMMON ERROR THAT A COMPUTERIST MAKES WITH THE NEXT-FOR ERROR IS TO NEST THE LOOPS IN THE WRONG ORDER. FOR INSTANCE, IF YOU HAVE A LINE WHICH SAYS 'IFX=1TO100:FORY=1TO100:NEXTX:NEXTY' YOU WOULD GET A 'NF' ERROR BECAUSE THE 'NEXT X' AND 'NEXT Y' ARE IN THE WRONG ORDER. COMPUTERISTS OFTEN FORGET THAT THE LAST 'FOR' HAS TO BE THE FIRST 'NEXT'.

UNFORTUNATELY, THE ERROR ALSO OCCURS WHEN YOU HAVEN'T DONE ANYTHING WRONG. BASIC HAS A LIMITED NUMBER OF LOCATIONS WHERE IT CAN STORE RETURN ADDRESSES FOR 'FOR-NEXT' LOOPS. IF YOU NEST TOO MANY LOOPS, BASIC WILL OVERWRITE THE FIRST ONES THAT WERE SET UP; THEN IF IT FINDS A 'NEXT' REFERRING TO THEM WON'T KNOW WHERE TO GO. LETS SEE IF WE CAN MAKE THAT CLEARER. IF YOU HAD A BASIC LINE THAT SAID 'FORA=1TO2:FORB=1TO3:FORC=1TO 5:FORD=10TO15:FORE=100TO110' AND ANOTHER LINE THAT SAID 'NEXTE: NEXTD:NEXTC:NEXTB:NEXTA' AND SO ON, SOONER OR LATER YOU WOULD GET TOO MANY FOR-NEXT LOOPS IN AND BASIC WOULD OVERWRITE THE A'S STORAGE LOCATION AND FORGET THAT IT WAS A PART OF A FOR-NEXT LOOP. IF IT THEN SAW A 'NEXT A' STATEMENT, IT WOULD REPORT AN 'NF' ERROR.

THERE IS ONE OTHER POSSIBLITY FOR A 'NEXT-FOR' MISTAKE. IT IS A COMMON ERROR, PARTICULARILY WITH NEW COMPUTERISTS, TO PUT IN MORE THAN ONE 'NEXT' STATEMENT FOR A GIVEN 'FOR'. IF QUESTIONED CLOSELY, I MIGHT EVEN ADMIT TO HAVING DONE IT MYSELF A FEW TIMES. THE ROUTINE USUALLY GOES LIKE THIS: 100 FORX=1TO10 110IFX=10THEN NEXT 120IFX<5THENNEXT 130...SOME MORE STUFF...:NEXT. YOU START OFF WITH THE IDEA THAT YOU ARE GOING TO JUMP BACK TO THE 'FOR' LOOP ON A NUMBER OF DIFFERENT OCCASSIONS. UNFORTUNATELY, IF THE LAST TIME THROUGH ONE OF THE FIRST 'NEXT'S IS EXECUTED, THEN WHEN IT HITS THE SECOND, AND FINAL, NEXT FOR THE SAME 'FOR' YOU WOULD GET A 'NF' ERROR. TO AVOID THAT, WHEN YOU ARE WRITING A PROGRAM REMEMBER THE HARD AND FAST RULE - YOU ARE ALLOWED ONLY ONE 'NEXT' FOR EVERY 'FOR'. IF YOU WANT TO INCREMENT THE COUNTER ON A NUMBER OF DIFFERENT OCCASIONS, THEN JUMP TO THE 'NEXT' STATEMENT RATHER THAN EXECUTING A 'NEXT' WHEREEVER YOU ARE.

THE NEXT THING TO COVER IS 'RG' - THAT'S 'RETURN WITHOUT GOSUB' THE USUAL PROBLEM WITH THAT IS THAT YOU HAVE SIMPLY FALLEN THROUGH A PART OF THE PROGRAM THAT YOU DIDN'T EXPECT TO GET INTO AND ALL OF A SUDDEN YOU ARE IN A SUBROUTINE THAT YOU HADN'T PLANNED TO REACH. FOR INSTANCE, IT IS COMMON TO PUT SUBROUTINES ON THE END OF A PROGRAM AND WE OFTEN FORGET TO PUT AN END STATEMENT ON THE MAIN PROGRAM. THEREFORE, OFTEN WHEN THE GAME IS OVER OR THE PROGRAM IS COMPLETED, SINCE

THERE IS NO END OR STOP STATEMENT ON IT, THE COMPUTER CONTINUES TO TRY TO EXECUTE THE PROGRAM AND FALLS INTO A SUBROUTINE WITHOUT BEING SENT TO IT. THAT COMES UP WITH THE 'RG' ERROR. THE SOLUTION IS TO LOOK AT THE END OF THE SUBROUTINE AND SEE LOOK FOR HOW YOU CAN GET INTO IT WITH OUT CALLING IT.

THE NEXT TOPIC IS THE 'BS' ERROR (NO, DIRTY MINDS, THAT IS NOT OBSCENE!). IT IS A VERY SIMPLE ERROR , BUT SOMETIMES VERY DIFFICULT TO ACTUALLY PIN DOWN. A 'BS' ERROR MEANS ONLY THAT YOU HAVE CALLED FOR AN ARRAY ELEMENT WHICH IS HIGHER THAN WHAT YOU DIMENSIONED FOR. FOR INSTANCE, IF YOU HAVE DIMENSIONED 'A' FOR 10 AND YOU TRY TO CALL B=A(11) THEN YOU GET A 'BS' ERROR. THERE ARE TWO POSSIBLE SOLUTIONS. IF YOU ACTUALLY DO WANT TO GET A(11) THEN GO BACK TO THE DIMENSION STATEMENT AND DIMENSION 'A' FOR A HIGHER NUMBER, OR AS OFTEN HAPPENS, YOU ADD A DIMENSION STATEMENT BECAUSE YOU FORGOT TO DO IT THE FIRST TIME. THE OTHER, AND MORE COMMON, PROBLEM IS THAT YOU HAVE CALCULATED WHICH VARIABLE YOU NEED AND MADE A MISTAKE IN THE CALCULATIONS. IN THAT CASE, YOU PRINT OUT THE STUFF INSIDE THE PARENTHESIS, FIND WHAT IT ADDS UP TO AND FIND OUT HOW IT GOT TO BE HIGHER THAN YOU PLANNED.

THE LAST COMMON AND EASY TO COPE WITH ERROR CODE IS 'OD' OR OUT OF DATA. THAT MEANS, OF COURSE, SIMPLY THAT YOU HAVE ASKED THE COMPUTER TO READ SOMETHING FROM DATA AND THERE ISN'T ANY MORE DATA TO READ. TROUBLESHOOTING THIS CAN BE A LITTLE IRRITATING, BUT IT IS ALWAYS SOMEWHAT SIMPLE. ALL THAT HAS HAPPENED IS THAT YOU HAVE A NUMBER OF ITEMS IN A DATA STRING AND YOU FORGOT TO TYPE ONE OF THEM IN. THE COMPUTER GOT TO THE END OF THE DATA STRING AND WASN'T AT THE END OF THE PROGRAM. ALL YOU HAVE TO FIGURE OUT IS HOW MANY DATA BYTES YOU ARE MISSING AND WHICH ONES THEY ARE. THE USUAL THING IN READING DATA STATEMENTS IS TO SET UP A FOR-NEXT LOOP. WE USE THEM FOR THINGS LIKE POKING IN MACHINE CODE SUBROUTINES, AND NORMALLY THERE IS AN 'FORX=SOMETHING TO SOMETHING:READ A DATA BYTE' STATEMENT. IF YOU GET AN 'OD' ERROR, LIST THE PROGRAM OUT TO THE POINT WHERE YOU GOT THE ERROR, TELL THE SYSTEM IN DIRECT MODE TO PRINT THE VALUE OF THE COUNTER AT THAT CURRENT TIME. FOR INSTANCE, IF YOU SAID 'FORX=1TO15:READA(X)' THEN TELL THE COMPUTER TO '?X'. THE VALUE OF X WILL TELL YOU HOW MANY DATA BYTES THERE ACTUALLY ARE THERE. BY SUBTRACTING THAT VALUE FROM HOW MANY YOU WANTED, IN THIS CASE 15, YOU KNOW HOW MANY YOU ARE MISSING. YOU CAN THEN EASILY ZERO IN ON WHICH ONES ARE MISSING IF YOU HAVE A LISTING OF WHAT THE DATA SHOULD BE. IN DIRECT MODE SIMPLY HAVE THE COMPUTER PRINT OUT TO YOU WHAT THE SELECTED DATA BYTES ARE UNTIL YOU FIND THE PROBLEM. FOR INSTANCE, IF YOU ARE READING IN FIFTEEN NAMES AND YOU HAVE A LIST OF WHAT THE FIFTEEN SHOULD BE, TELL THE COMPUTER TO '?A$(7)' AND SEE IF THAT IS CORRECT. IF IT ISN'T, THE ERROR OCCURED BEFORE THE SEVENTH NAME AND IF IT IS THE ERROR IS AFTER SEVEN. YOU THEN CAN FIND WHICH BYTES ARE ACTUALLY MISSING IN THE SAME WAY. IT CAN BE SLIGHTLY TEDIOUS BUT NOT REALLY DIFFICULT OR BRAIN-BUSTING NORMALLY. EXCEPT FOR THE OUT OF MEMORY ERROR, YOU HAVE TO WORK HARD TO GET ANY OF THE OTHER ERROR CODES TO APPEAR AND WITH LUCK YOU'LL NEVER SEE ONE. THE OUT OF MEMORY ERRORS YOU SHOULD BE ABLE TO HANDLE FROM THE ARTICLES WE RAN IN ISSUE #4 ON HOW TO SAVE MEMORY IN A PROGRAM, BUT WE CAN REPEAT A FEW OF THE QUICK FIX TYPE SOLUTIONS IN THIS ARTICLE. IF YOU GET AN 'OM' ERROR THE FIRST THING TO DO IS TO GET RID OF THE REMARKS YOU SEE IN THE PROGRAM, SHORTEN THE PRINT STATEMENTS, GET RID OF THE INSTRUCTIONS, AND DELETE AS MANY BYTES IN LINE NUMBERS AS CAN OBVIOUSLY BE DONE FROM FIRST EXAMINATION. THOSE TRICKS PRETTY MUCH COMPRISE A FIRST AID/BAND AID BOX FOR 'OM' ERRORS. IF YOU STILL HAVE AN 'OM' ERROR AFTER YOU DO THOSE THINGS, THEN YOU HAVE TO GET INTO REWRITING THE PROGRAM ALONG THE LINES DISCUSSED PREVIOUSLY IN THE JOURNAL.

TROUBLESHOOTING TECHNIQUES

THE REALLY NEAT THING ABOUT BASIC THAT MAKES IT POSSIBLE TO TROUBLESHOOT BOTH THE TYPE OF ERROR WHICH WE HAVE ALREADY DISCUSSED AND THE MORE TROUBLESOME LOGIC ERROR WHICH WE ARE NOW GOING TO GET INTO IS THAT IT IS AN INTERPRETED RATHER THAN A COMPILED LANGUAGE. WHAT THAT MEANS IN END USER TERMS IS THAT YOU CAN MODIFY, PRINT OUT FROM, AND EXAMINE THE PROGRAM WHILE IT IS RUNNING. YOU CAN EITHER STOP IT AND ASK FOR INFORMATION OR ASK IT TO CONTINUOUSLY PRINT OUT INFORMATION AS IT EXECUTES. THE TECHNIQUES ARE THESE:

THE FIRST, AND SIMPLIEST, TECHNIQUE, ASSUMING THAT YOU HAE NOT TURNED OFF THE CONTROL C, IS TO BREAK THE PROGRAM WITH CONTROL C AND ASK THE COMPUTER TO PRINT OUT A VARIABLE VALUE. FOR INSTANCE, IF YOU ARE GETTING A FUNCTION CALL ERROR IN A LINE AND YOU ARE NOT CERTAIN WHERE THE OUT OF RANGE VALUE IS COMING FROM, YOU CA BREAK THE PROGRAM SEVERAL TIMES, AND ASK FOR THE

VARIABLE TO BE PRINTED SO YOU CAN SEE WHERE IT IS AT ANY ONE MOMENT. AFTER TYPING CONTROL C AND USING A DIRECT COMMAND TO PRINT OUT WHATEVER INFORMATION YOU WANT, YOU CAN CONTINUE THE PROGRAM WITH A 'CONT' COMMAND.

THE NEXT SIMPLE TECHNIQUE IS TO HAVE THE COMPUTER PRINT OUT THE VARIABLES AS THE PROGRAM EXECUTES. FOR INSTANCE, IN A RECENT GAME THAT I WROTE, I WAS UNABLE TO DETERMINE FROM EXAMINING THE PROGRAM WHY ONE OF THE KEYS WAS NOT FUNCTIONING AS PREDICTED. THE CONTROL DIDN'T WORK AND I COULDN'T FIGURE OUT WHY NOT FROM LOOKING. I THEREFORE HAD THE COMPUTER CONTINUOUSLY PRINT OUT THE VALUES RECEIVED FROM PEEKING THE KEYBOARD CONTINUOUSLY AND PRESSED THE KEY. THAT TOLD ME IMMEDIATELY, IN REAL TIME TERMS, WHAT VALUE THE COMPUTER WAS GETTING FROM THE KEY.

THERE IS EVEN A NEAT TRICK FOR DOING THAT WITHOUT DISTURBING THE REST OF YOUR PROGRAM. IF FOR INSTANCE, YOU ARE TROUBLESHOOTING A VIDEO GAME OR A BUSINESS PROGRAM WITH LOTS OF GRAPHICS ON THE SCREEN, AND DO NOT WANT TO DISTURB THOSE WHILE THE PROGRAM RUNS YOU CAN HAVE INFORMATION PRINTED OUT ON A REAL-TIME BASIS ANYWAY. WHAT YOU DO IS TO USE THE CHR$(13) COMMAND. THE CONTROL CHARACTER THIRTEEN TELLS THE SYSTEM TO DO A CARRIAGE RETURN BEFORE IT PRINTS, BUT NOT TO SCROLL THE SCREEN. TO KEEP IT FROM SCROLLING AFTER THE PRINT, YOU END THE PRINT STATEMENT WITH A SEMI-COLON (;). FOR INSTANCE, WHEN I WAS PEEKING THE KEYBOARD IN THAT LAST EXAMPLE, WHAT I ACTUALLY PUT INTO THE COMPUTER WAS "601?CHR$(13)PEEK(57100);". THAT PRINTED THE KEYBOARD PEEK EVERYTIME THE PROGRAM WENT THROUGH THAT SECTION AND SINCE IT DID A CARRIAGE RETURN BEFORE IT PRINTED IT AND DID NOT SCROLL AFTERWARDS, IT CONTINUALLY PRINTED THE VALUE IN THE LOWER LEFT HAND CORNER OF THE SCREEN. IT IS A PARTICULARLY HANDY TROUBLE SHOOTING TECHNIQUE.

A SIMILAR TRICK CAN BE USED FOR A QUICK AND DIRTY TRACE FUNCTION. FOR THOSE OF YOU WHO ARE NOT FAMILIAR WITH IT, A TRACE PROGRAM PRINTS TO THE SCREEN THE LINE NUMBER OF EACH LINE AS IT IS EXECUTED. THAT WAY YOU CAN SEE WHERE THE PROGRAM GOES AT EACH BRANCH. TRACE PROGRAMS ARE AVAILABLE FOR THE OSI, (AARDVARK SELLS ONE) BUT THEY HAVE THE DISADVANTAGE THAT THEY HAVE TO BE LOADED INTO THE MACHINE EVERYTIME YOU WANT TO USE IT. IF YOU ARE IN THE MIDDLE OF TROUBLE SHOOTING A PROGRAM, IT IS OFTEN EASIER TO USE A TRICK RATHER THAN A TRACE. ASSUME THAT YOU ARE HITTING A 'RG' ERROR AND YOU DON'T KNOW FROM LOOKING AT THE PROGRAM LISTING HOW IT IS GETTING INTO THAT SUBROUTINE, YOU CAN CHECK BACK ALONG THE LINE NUMBERS TO SEE WHICH ONE IS EXECUTED JUST BEFORE THE 'RG' ERROR AND THEREFORE DETERMINE WHERE THE PROGRAM IS JUMPING INTO THAT SUBROUTINE. YOU DO IT USING THE CHR$ FUNCTION AND NUMBERS. IF YOU HAVE A 'RG' ERROR IN LINE 600 AND YOUR PROGRAM IS NUMBERED BY EVEN 10'S, YOU CAN GO BACK SEVERAL LINES AND INSERT LINE NUMBERS LIKE THESE: 561 ?CHR$(13)561 581 ?CHR$(13)581 AND SO ON, SO THAT AS THE PROGRAM EXECUTES THOSE LINES IT WILL PRINT IN THE LOWER LEFT HAND SCREEN THE NUMBER SHOWING THAT IT HAS JUST GONE THROUGH THAT LINE. THAT CAN SOMETIMES QUICKLY AND EASILY TELL YOU WHERE A PROGRAM IS GOING WRONG.

OF COURSE, THERE IS ALWAYS OUR OLD STANDBY OF THE STOP STATEMENT. IT IS THE RECOMMENDED 'BY THE BOOKS' METHOD OF TROUBLE SHOOTING BASIC PROGRAMS. YOU CAN LIBERALLY SPRINKLE STOP STATEMENTS THROUGHOUT THE PROGRAM AND THEN WHEN YOU RUN IT IT WILL BREAK AT THOSE POINTS. YOU CAN THEN DO A NUMBER OF DIFFERENT THINGS OR CONTINUE THE PROGRAM WITH A 'CONT' STATEMENT. YOU CAN, FOR INSTANCE, TELL THE COMPUTER TO PRINT OUT A VARIABLE VALUE, I.E. ?X. YOU CAN CHANGE A VARIABLE VALUE BY DIRECT COMMAND, I.E. YOU CAN TYPE IN X=31 OR A$="U". YOU CAN ALSO LIST OUT PART OR ALL OF A PROGRAM. WHEN YOU ARE DONE, YOU TYPE IN 'CONT' AND CONTINUE WITH THE PROGRAM. THERE ARE A COUPLE OF THINGS YOU MUST NOT DO IF THIS IS TO WORK. YOU MAY NOT CHANGE ANY BASIC LINE, DOING SO CAUSES ALL THE VARIABLES TO BE CLEARED. YOU MAY NOT CAUSE A SYNTAX ERROR AS THE SYSTEM WILL THEN REFUSE TO CONTINUE.

THE ONE FINAL COMMENT IS THAT YOU SHOULD NOT BE AFRAID TO MANIPULATE THE PROGRAM WITH DIRECT COMMANDS. FOR INSTANCE, IF THE PROGRAM MALFUNCTIONS ONLY WHEN THE SCORE OF THE FIRST PLAYER IS OVER TWENTY, DON'T BE AFRAID TO STOP THE PROGRAM, TYPE IN SCORE = 21 AND CONTINUE. THERE IS NO POINT WAITING FOR THE MACHINE TO GO THROUGH A LONG CYCLE TO REACH THE AREA OF VALUES THAT YOU BELIEVE CAUSE THE MALFUNCTION. SIMPLY STOP THE MACHINE, ASSIGN THE VALUES AND GO ON.

THE SAME COURAGE AND FORTHRIGHTNESS (?) SHOULD BE SHOWN WITH A PROGRAM FLOW PROBLEM. I ONCE WATCHED A PROGRAMMER TRY FOR HALF AN HOUR TO TROUBLE SHOOT A PROGRAM THAT ONLY FOULED UP WHEN THE SCORE WAS OVER 100 AND THE PROGRAM WENT TO A BONUS ROUTINE. HE KEPT PLAYING THE GAME TRYING TO GET A BONUS SO THAT THE PROGRAM WOULD GO TO THE PROBLEM SUBROUTINE. YUCHCHCHCH!. WHEN YOU HAVE A PROGRAM THAT IS FOULING UP ONLY IN CERTAIN AREAS, BREAK THE PROGRAM, TYPE IT OVER WITH. YOU DON'T HAVE TO COAX A PROGRAM TO GO SOMEWHERE, YOU JUST TELL IT.

RG

** ADVERTISEMENTS **

C24P 8K. WORKS WELL $450. GREG PHILLIPS 7860 AUTUMN DR. BEAMONT TEXAS. 77706. (713) 866-5887
C1P8K SUPERMOD II. SHIFT LOCK KEY WON'T ALWAYS STAY DOWN BUT IS REPAIRABLE. MUST BE ABLE TO PICK UP. 8 MO. OLD. $400 (312) 366 0276 RANDOLPH R. MADERA. 214 CIRCLE APT 202 , FOREST PARK, ILL. 60130
C3 HARD DISK, HAZELTINE TERMINAL, LINE PRINTER, ACCOUNTS RECIEVABLE SOFTWARE. $14,500. JERRY GOTLIEB 3247 LAKEWOOD ANN ARBOR MI. 48103

THE BEGINNER'S CORNER

THE VENERABLE 'ON...GOTO...' STATEMENT.

THE ON..GOTO STATEMENT IS ONE OF THE NEATEST TIME AND MEMORY SAVERS IN BASIC. BUT IT IS UNFORTUNATELY RATHER POORLY DOCUMENTED IN MOST BASIC TEXTS. THERE ARE SOME HANDY BUILT IN TRICK TO IT THAT ALLOW IT TO BE MORE USEFUL THAN IT APPEARS AT FIRST.

THE FIRST THING TO REMEMBER WHEN YOU WANT TO USE AN 'ON...GOTO..' STATEMENT IS THAT YOU CAN USE AN EXPRESSION IN PLACE OF A VARIABLE NAME FOR THE ON..GOTO.. YOU CAN, FOR INSTANCE, TYPE IN 'ONX/100GOTO...' AND GIVE IT A NUMBER OF LINE NUMBERS. IT IS THEREFORE POSSIBLE TO PRECONDITION THE VARIABLES TO MAKE THEM FIT INTO AN ON..GOTO.. STATEMENT. IF YOU WANT TO DO A BUNCH OF JUMPS BASED ON THE LETTERS 'A' THROUGH 'D', YOU CAN TAKE THE ASCII OF THE LETTER AND JUMP ON THE ASCII. YOU CAN SAY 'ON(ASC)A$-64 ,GOTO100,200,300,400. IT WOULD THEN TAKE THE ASCII OF A$ WHICH IN THE CASE OF A,B,C,&D WOULD BE 65,66,67,68, SUBTRACT 64 FROM THOSE VALUES AND DO THE JUMPS ON THE BASIS OF WHAT IS LEFT OVER.

YOU SHOULD ALSO REMEMBER THAT THE NUMBER DOESN'T HAVE TO BE AN INTERGER. IN AN ON..GOTO.. STATEMENT, THE SYSTEM WILL JUMP TO THE FIRST NUMBER IF THE VARIABLE IS ANYTHING FROM 1 TO 1.999999 AND THE SECOND IF IT IS ANYTHING FROM 2 TO 2.99999. THE INTERGER FUNCTION DOESN'T HAVE TO BE SPECIFIED, IT SIMPLY OCCURS WITH THE EXECUTION OF THE LINE. THAT MAKES IT POSSIBLE TO DO A JUMP BASED ON A RANGE OF THINGS. IF YOU WANT TO GO TO ONE BONUS SEGEMENT IF THE SCORE IS 100 TO 199 AND ANOTHER IF SEGEMENT IF THE SCORE IS 200 TO 299, ALL YOU HAVE TO DO IS SAY ONX(OR SCORE)/100GOTO100,200. THE SYSTEM WOULD DIVIDE THE SCORES UP, GET A NUMBER BETWEEN ONE AND THREE, AUTOMATICALLY TAKE THE INTERGER OF IT AND JUMP ON THE INTERGER. THAT CAN SAVE YOU A COUPLE PROGRAM STEPS.

THE THIRD THING TO REMEMBER ABOUT USING AN ON..GOTO.. STATEMENT IS THAT IT IS HANDY TO USE THE FALL THROUGHS. IF THE VARIABLE HAS A VALUE OF ZERO OR A VALUE GREATER THAN THE NUMBER OF LINE NUMBERS STATED AFTER IT, IT SIMPLY IGNORES THE ON..GOTO.. STATEMENT AND FALL THROUGH. THEREFORE, IF YOU HAVE A SYSTEM WHERE YOU WANT TO GO SOMEWHERE BASED ON SAY A SCORE OF 1, 2, OR 3 AND GO ELSEWHERE IF THE SCORE IS ZERO OR OUT OF RANGE BEING FOUR OR MORE, YOU COULD SAY ONXGOTO100,200,300:GOTO500. WITH A LINE LIKE THAT IF THE X IS WITHIN THE RANGE YOU SPECIFY, IT WILL JUMP TO ONE OF THE THREE LINES. WITH ANY OTHER POSITIVE VALUE IT WILL THEN GO TO 500.

THE ONLY THING THAT YOU REALLY HAVE TO WATCH FOR WHICH CAN CAUSE REAL PROBLEMS USING THE ON..GOTO.. STATEMENT IS THAT YOU MUST NOT HAVE A NEGATIVE NUMBER AS THE VARIABLE. ANY NEGATIVE NUMBER WILL CAUSE A FUNCTION CALL ERROR.

AT LAST!!! VIDEO MODS FROM AARDVARK!!!

***** CHEAP AND SIMPLE TO INSTALL. NO MORE BUTCHER STUFF.*****

I TRY TO AVOID HAWKING AARDVARK STUFF IN THE JOURNAL BUT THIS IS A REAL SORE POINT WITH ME. I FEEL THAT A LOT OF PEOPLE HAVE BEEN RIPPED OFF WITH $20 SETS OF UNWORKABLE PLANS. IN THE PAST WE HAVE TRIED TO HELP BY OFFERING FREE SETS OF THE PLANS TO ALL WHO WANTED THEM. THEY STILL WEREN'T PRACTICAL BUT AT LEAST YOU WEREN'T HIT FOR $20.00 FOR THEM.

NOW WE ARE GOING TO GO ONE BETTER. THE UK USERS GROUP SENT US A WORKABLE SCHEME FOR A 32X32 DISPLAY. IT IS A GOOD MOD WHICH ABOUT DOUBLES THE NUMBER OF CHARACTERS ON THE SCREEN (FROM 576 TO 1024) AND IT CAN BE DONE WITH THREE CHIPS AND A COUPLE OF CHEAP SUPPORT PARTS.

HERES THE REALLY GOOD NEWS. THE SCHEMATIC IS IN THIS JOURNAL - FREE - .

OH - ABOUT THE AARVARK PRODUCTS. WE ARE OFFERING TWO DATA SHEETS DETAILING SIMILAR MODS WITH

INSTRUCTIONS THAT WOULD GET A 12 YEAR OLD THROUGH IT WITHOUT A HITCH.

V-MOD 1 IS A PLAN SIMILAR TO THE FREE ONE HERE. IT GETS RID OF THE OVERSCAN WITH ABOUT THE SAME SCHEME. IT INCLUDES PLANS FOR BREADBOARD THE CIRCUIT ON A SEPARATE CARD AND TYING IT IN. THE MOD IS SIMPLE AND SHOULD COST LESS THAN $15.00 TO INSTALL. I HAVE TO POINT OUT THAT IT DOES HAVE SOME LIMITATIONS. AS IT IS NOT CRYSTAL CONTROLLED, IT HAS TO BE CAREFULLY ADJUSTED WHEN FIRST INSTALLED (A TEETH GRITTING 10 TO 15 MINUTES THE FIRST TIME YOU TRY IT) AND IT MAY NOT BE PARTICULARLY PORTABLE. ONCE SET UP FOR YOUR MONITOR, YOU ARE FINE, BUT IF YOU CHANGE MONITORS, YOU MAY HAVE TO READJUST THE POTS. - STILL, A GOOD CHEAP SIMPLE MOD.

VIDEO MOD 2 IS A LITTLE MORE COMPLEX ONLY IN THAT YOU HAVE TO INSTALL A NEW CRYSTAL (A CHEAP AND COMMON $3.00 TYPE) AND DO ABOUT TWO MORE JUMPERS. IT IS, HOWEVER, SUPER STABLE AND VERY PROFESSIONAL. IT HAS THE ADVANTAGE OF UPPING YOUR CPU CLOCK RATE TO 1.4 MEG, MAKING GAMES NICER, AND ENABLING YOU TO LOAD TAPE AT 600 AND 1200 BAUD. - NOT BAD. THE PLANS - (PLANS - PHOOEY- IT IS BETTER DESCRIBED AS A SET OF INSTRUCTIONS HEATH KIT WOULD BE PROUD OF.) COST $7.95 FROM AARDVARK.

FOR THOSE OF YOU WHO ARE TECHNICALLY ORIENTED, THE SCHENATICS PUBLISHED HERE SHOULD BE ADEQUATE.

VIDEO DISPLAY MOD

(KILLNG 'OVERSCAN' ON C1/SUPERBOARD)

HERE IS A SIMPLE WAY OF ACHIEVING THE ASPIRATION OF MANY SUPERBOARD OWNERS - THAT OF GETTING RID OF THE 'OVERSCAN' LIMITATION TO THE NUMBER OF CHARACTERS PER TV LINE. THE SOLUTION, WHILE NOT ELEGANT, IS EXTREMELY SIMPLE, USING THREE OR FOUR CHIPS AND A FEW PASSIVE COMPONENTS. THE METHOD WILL BE DESCRIBED ONLY IN GENERAL TERMS, ALLOWING FOR ALL THE VARIANTS IN SYSTEMS, BUT ANYONE WHO BASICALLY UNDERSTANDS THE GUTS OF THEIR MACHINE WILL BE ABLE TO IMPLEMENT THE MOD IN A FEW EVENINGS. THE QUESTION OF USING THE MOD TO ITS FULL ADVANTAGE IS A BIT TRICKIER. RAM BASED OPERATING CAN USE THE FULL LINE LENGTH; OTHERS WITH ROM SYSTEMS WILL ONY BE ABLE TO USE THE EXPANDED SCREEN THROUGH POKES.

THE CRUCIAL ASPECT OF THE SOLUTION IS THE PROVISION OF TWO CLOCKS: ONE FOR THE PROCESSOR AND CASSETTE INTERFACE, THE OTHER FOR VIDEO. INSPECTION OF THE CIRCUIT DIAGRAM SHOWS THAT THE FIRST PART NEEDS ONLY TWO SIGNALS - THE $\emptyset_0$-IN AND THE INPUT TO THE TXCLK SYSTEM (U57). THESE ARE READILY PROVIDED BY TAKING THE 4MHZ OUTPUT OF THE CRYSTAL SYSTEM (PIN 3, U58) AND PUTTING IT THROUGH A 74163 (OR DIVIDING CHAIN EQUIVALENT TO IT) WHICH WILL PROVIDE 2MHZ OR 1MHZ SIGNALS FOR THE PROCESSOR AND A SIGNAL THAT WILL DIVIDE DOWN TO THE EQUIVALENT OF 600 BAUD FOR THE CASSETTE SYSTEM. IF 300 BAUD OPERATION IS NEEDED THEN A FURTHER DIVIDE-BY-TWO STAGE (I.E. A 7474) IS SIMPLY ADDED.

NOW FOR THE VIDEO PART. THIS ALL USES THE CLK SIGNAL - I.E. ONLY ONE INPUT IS REQUIRED TO OPERATE IT ALL. ASSUMING WE HAVE CUT THE CLK TRACE WITH THE CRYSTAL CONTROLLING ONLY THE PROCESSOR AND CASSETTE, WE NEED ONLY ONE OTHER OSCILLATOR TO DRIVE THE WHOLE VIDEO SYSTEM. IN PRINCIPLE, ANOTHER QUARTZ CRYSTAL WOULD SEEM DESIRABLE, BUT INSTEAD WE PROVIDE AN ADJUSTABLE CLOCK USING A SCHMITT TRIGGER 74132 GATE WITH A CONTROLLABLE RC NETWORD. WE CAN NOW GET THE CHARACTERS COMING OUT FASTER THAN THE OLD SYSTEM BY TAKING THE FREQUENCY ABOVE 4MHZ, AND HENCE GET MORE CHARACTERS PER LINE. BUT THERE'S ONE THING MISSING, THE CHARACTERS ARE STILL BEING SPEWED OUT IN THE OVERSCAN REGION. TO SOLVE THIS PROBLEM, ALL WE NEED DO IS TO STOP THE CLOCK WHEN YOU'RE OVERSCANNING! THIS IS VERY SIMPLE TO DO: WHEN THE HORIZONTAL SYNC PULSE COMES ALONG WE INITIATE A DELAY EQUAL TO THE OVERSCAN TIME. THE DELAY PULSE GATES THE CLOCK (USING THE SAME '132 GATE CHIP AS PROVIDES THE CLOCK) AND ALL VIDEO ADDRESSING STOPS. BECAUSE THE VIDEO SYSTEM IS COMPLETELY ISOLATED FROM THE PROCESSOR SYSTEM, ALL THIS STOPPING AND STARTING OF THE VIDEO CLOCK AT ITS OWN ODD FREQUENCY HAS NO EFFECT ON THE PROCESSOR OR CASSETTE.

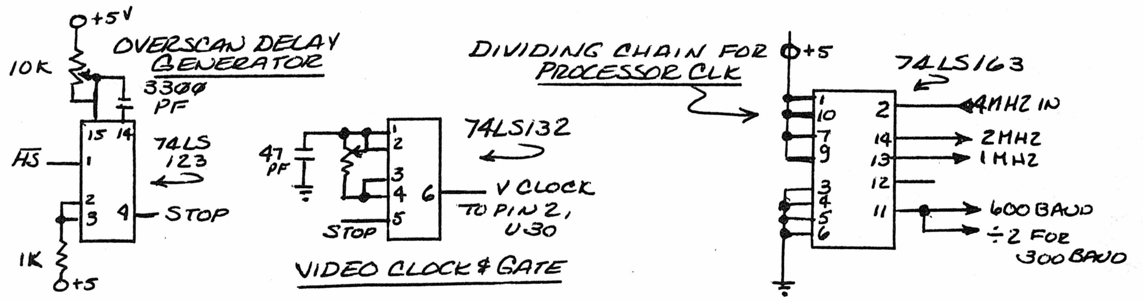
IT MAY SOUND FROM THE DESCRIPTION THAT TO SET ALL THIS UP WOULD REQUIRE SOPHISTICATED SCOPES AND TIMING GEAR! IN FACT, ALL YOU NEED TO DO IS GET YOUR RESISTORS AND CAPACITORS IN THE RIGHT SORT OF AREA AND THEN PLAY WITH THE RESISTANCES UNTIL YOU GET THE RIGHT COMBININATION OF DELAY (SO YOU LOSE NO CHARACTERS TO THE LEFT OF THE SCREEN) AND OF SPEED OF VIDEO OUTPUT (SO YOU LOSE NONE TO THE RIGHT). SURPRISINGLY THE DELAY IS VERY STABLE, EVEN WITHOUT CRYSTAL CONTROL, PROVIDED YOU USE DECENT TRIM POTS. INITIAL VERSIONS WITH UNTIDY WIRING SHOWED SOME INTERFACE BETWEN VARIOUS SIGNALS, BUT AS SOON AS THE WIRING WAS TIDIED UP, EVERYTHING BEHAVED ITSELF. THE

DISPLAY 'LOCKS IN' ON ALL TVS, THOUGH, OF COURSE, MINOR ADJUSTMENTS HAVE TO BE MADE TO SUIT THE DIFFERING OVERSCANS OF THE PARTICULAR MODELS.

NOTES: 1) 74123 IS USED BECAUSE IT'S FAMILIAR, A 74121 WOULD DO AS WELL. THE SPARE MONOSTABLE ON THE '123 MIGHT ALWAYS COME IN HANDY.

2) RESISTANCES AND CAPACITANCE FIT MY SUPERBOARD WITH THE RESISTANCES IN THE MIDDLE OF THEIR ADJUSTABLE RANGES.

3) 'STOP' CAN ALSO BE CONNECTED TO PIN 10 OF U56 TO ACTIVATE (DB) - THIS TIDIES UP THE SCREEN DURING SYNC AND DELAY.



RELOCATING THE EXTENDED MONITOR (EXMON)

SINCE THE STANDARD VERSION OF THE EXTENDED MONITOR IS LOCATED AT THE END OF THE FIRST 4K, THIS MEANS THAT IT IS LOCATED RIGHT IN THE MIDDLE OF RAM ON AN 8K MACHINE, AND, INCIDENTLY, ALSO IN THE MIDDLE OF THE ASSEMBLER. EXMON DOES HAVE A RELOCATE FUNCTION OF ITS OWN, WHICH CORRECTS ALL SUBROUTINE CALLS AND JUMP ADDRESSES, BUT LOOK-UP TABLES AND THE LIKE ARE EITHER SCRAMBLED (IF THEY APPEAR TO BE JUMPS OR JSRS) OR LEFT UNTOUCHED. TO MOVE EXMON TO ANYWHERE ELSE IN MEMORY, ITS JUMP-TABLE MUST BE CHANGED BY HAND AFTER USING THE RELOCATE - THE JUMP TABLE RESIDES AT $0960-0999 IN THE STANDARD VERSION. THE JUMPS ARE IN PAIRS, WITH THE LOW BYTE FIRST, AS USUAL. TO MOVE EXMON TO THE TOP OF AN 8K MEMORY, FOR EXAMPLE, $10 MUST BE ADDED TO EVERY HIGH BYTE - THE CONTENTS OF $0961, 0963, 0965, AND SO ON. $0962-0963 ARE THE ADDRESS FOR EXMON'S 'A' ROUTINE (PRINT CONTENTS OF ACCUMULATOR), $0964-0965 ARE THE ADDRESS OF 'B', AND SO ON TO 'Z'.

MICHAEL WHITTLE INCLUDED SEVERAL OTHER COMMENTS ON EXMON IN HIS LETTER. ONE WAS A COMPLETE PATCH TO ALLOW EXMON TO 'SAVE' IN THE ROM MONITOR'S HEX-DIGIT FORMAT RATHER THAN THE STRANGELY UNRELIABLE CHECKSUM - THE LISTING IS BELOW. THIS DUMP ROUTINE IS SHORTER THAN THE CHECKSUM DUMPER, SO - AS HE SAYS - 'THERE IS ROOM FOR ANOTHER GOODY', NAMELY A ROUTINE TO RESTORE THE VITAL (FOR BASIC) ADDRESSES $00D1-D6, AND HAVING DONE SO, TO JUMP TO BASIC. IN ORDER TO IMPLEMENT THIS ROUTINE $0994-5 (OR THEIR RELOCATED EQUIVALENTS) NEED TO BE SET TO POINT TO THE ROUTINE'S START ADDRESS, SO THAT THE SPARE 'Z' COMMAND WILL IMPLEMENT THE JUMP FROM EXMON TO BASIC. FURTHER MONITOR ENHANCEMENTS ARE TO USE SPACE INSTEAD OF CTRL-J TO INCREMENT TO THE NEXT LINE IN '@' AND 'Q' MODES. THE CHANGE IS ACHIEVED BY CHANGING $0A TO $20 AT (STANDARD) LOCATIONS $0B70 AND $0D2F. IT IS ALSO HELPFUL, SAYS MICHAEL, TO MAKE THE QUOTES (FOR ASCII) ADVANCE TO THE NEXT LINE, ENABLING AN ASCII STRING TO BE LISTED RAPIDLY. THIS IS ACCHIEVED BY CHANGING $0B81 FROM $60 TO $8B.

LISTING TO CHANGE CHECKSUM TO DIGIT-PAIR SAVE - TOP OF 8K MEMORY

```
1EC3   207FFF   JSR $FFF7 ;'S'ENTRY POINT, SET SAVE FLAG
1EC6   201C1B   JSR $1B1C ; GET START & STOP ADDRESS. STORE IN DC-DF
1EC9   A93A     LDA  $3A  ;':' PROMPT FOR JUMP ADDRESS
1ECB   2069FF   JSR $FF69 ;OUTPUT ROUTINE-ADJUST ON C2
1ECE   20A31A   JSR $1AA3 ;GET BYTE-ADDRESS HIGH
1ED1   85C1     STA $C1 ;STORE IN C1
1ED3   20A31A   JSR $1AA3 ;GET ADDRESS LOW
1ED6   85C0     STA $C0   ;STORE IN C0
1ED8   A92E     LDA  $23;'.'FOR ADDRESS MODE 1EDA    2069FF   JSR $FF69 ;OUTPUT
1EDD   A5DD     LDA $DD   ;START ADDRESS-HIGH BYTE
```

```
1EDF   20AC1A   JSR $1AAC ;GUTPUT AS HEX PAIR
1EE2   A5DC     LDA $DC   ;START ADDRESS-LOW BYTE
1EE4   20AC1A   JSR $1AAC ;OUTPUT AS HEX PAIR
1EE7   A92F     LDA $2F   ;'/' FOR DATA MODE
1EE9   2069FF   JSR %FF69 ;OUTPUT
1EEC   A200     LDX $00   ;CLEAR X REGISTER AS POINTER
1EEE   A1DC     LDA ($DC,X);GET NEXT DATA BYTE
1EF0   20AC1A   JSR $1AAC ;OUTPUT DATA BYTE AS HEX PAIR
1EFC   A90D     LDA $0D   ;'CR'
1EF5   20B1FC   JSR $FCB1 ;OUTPUT CR TO CASSETTE ONLY -JSR $BF15 ON C2
1EF8   E6DC     INC $DC   ;INCREMENT LOW BYTE OF ADDRESS POINTER
1EFA   D002     BNE $1EFE ;SKIP NEXT INSTRUCTION IF LOW BYTE NOT ZERO
1EFC   E6DD      INC $DD   ;INCREM. HIGH BYTE OF ADDRESS IF LOW BYTE ZERO
1EFE   A5DE      LDA $DE   ;GET END ADDRESS, LOW BYTE
1F00   C5DC      CMP $DC   ;COMPARE AGAINST CURRENT ADDRESS POINTER LOW BYTE
1F02   D0E8      BNE $1EEC ;LOOP BACK FOR NEXT DATA BYTE IF NO MATCH
1F04   A5DF      LDA $DF   ;GET HIGH BYTE OF END ADDRESS
1F06   C5DD      CMP $DD   ;COMPARE WITH CURRENT ADDRESS POINTER, HIGH BYTE
1F08   D0E2      BNE $1EEC ;LOOP BACK FOR NEXT DATA BYTE IF NO MATCH
1F0A   A92E      LDA  $2E  ;'.' FOR ADDRESS MODE (FOR RESTART/JUMP ADDRESS)
1F0C   2069FF   JSR $FF69  ;OUTPUT
1F0F   A5C1     LDA $C1    ;GET RESTART/JUMP ADDRESS, HIGH BYTE
1F11   20AC1A   JSR $1AAC  ;OUTPUT AS HEX PAIR
1F14   A5C0     LDA $C0    ;GET RESTART/JUMP ADDRESS, LOW BYTE
1F16   20AC1A   JSR $1AAC  ;OUTPUT AS HEX PAIR
1F19   A947     LDA  $47   ;'G' FOR ROM MONITOR 'GO' COMMAND
1F1B   2069FF   JSR $FF69  ;OUTPUT
1F1E   A900     LDA  $00   ;GET NULL
1F20   8D0502   STA $0205  ;CLEAR SAVE FLAG
1F23   4C0918   JMP $1809  ;END-JUMP TO EXMON WARM START
1F26   A206     LDX  $06   ;'Z' ENTRY POINT-SET COUNTER FOR 6 BYTES
1F28   BD321F   LDA $1F32,X;LOAD DATA SAVED BEFORE OVERWRITE BY DISASSEMBLER.
1F2B   95D0     STA $D0,X  ;RESTORE TO D1-D6
1F2D   CA       DEX        ;DECREMENT BYTE COUNTER
1F2E   D0F8     BNE $1F28  ;LOOP BACK UNTIL ALL RESTORED
1F30   2074A2   JSR $A274  ;JUMP TO BASIC WARM START
1F33   E9D0     SBC  $D0   ;DATA ONLY-TAIL END OF BASIC'S 00BC SUBROUTINE
1F35   60       RTS        ;DATA
1F36   80       ???        ;DATA
1F37   4F       ???        ;DATA
1F38   C7       ???        ;DATA
1F39   EA       NOP        ;
1F3A   EA       NOP        ;PAD TO START EXMON 'V' ROUTINE AT 1F3B
```

THE PRACTICAL COMPUTER

IN ORDER TO BE PRACTICAL, THE COMPUTER DOES NOT HAVE TO COPE WITH HORRENDOUSLY COMPLICATED PROGRAMS. WE'RE IN THE MARKET FOR A NEW CAR AS MY TRUSTY BUG AND MICHIGAN WINTERS DO NOT GET ALONG SO WELL. I LIKE SMALL, RODGER LIKES BIG. IN AN EFFORT TO CONVINCE HIM THAT A NEWER SMALL GAS EFFICIENT CAR WOULD SAVE MONEY IN THE LONG RUN OVER AN OLDER, CHEAPER GAS-HOG, I CAME UP WITH THE PROGRAM ESTIMATING THE COST OF GAS FOR 10,000 MILES OF DRIVING PER YEAR FOR 10 YEARS (SO I'M OPTIMISTIC THAT A NEW CAR WILL EQUAL THE DURIBILITY OF A VW BUG....). THE ORIGNAL PROGRAM ALLOWED THE MPG RATING OF THE CAR TO BE ENTERED AND THEN COMPUTED THE COST OF 100,000 MILES AT A SET COST OF GAS ($1.20 - THE CURRENT COST AT THE CORNER STATION). SO FAR, SO GOOD - THE 28MPG SUB COMPACT WOULD ALMOST PAY FOR ITSELF OVER THE TEN YEAR PERIOD. BUT GAS ISN'T GOING TO STAY AT $1.20 A GALLON NO MATTER HOW MUCH WE WISH IT WOULD - SO I ADDED A INFLATION FACTOR

WHICH WOULD ALLOW INPUTING THE ESTIMATED RATE OF INFLATION AND WATCHED THOSE SAVINGS CLIMB.

NOW THAT I HAVE HIM CONVINCED THAT SMALL IS BEAUTIFUL, ALL I HAVE TO DO IS FIND A SMALL CAR HE CAN FIT INTO WITHOUT DECAPITATING HIM .... 6' 2 1/2" DOES NOT AGREE WITH LOW CEILINGS IN THE CARS THAT CAME OUT BEST IN GAS SAVINGS!!

```
 110 INPUT"MILES PER GALLON";M
 120 INPUT"PRICE OF GAS IN DOLLARS";G
 130 INPUT"INFLATION RATE (# OF %)";I:I=I/100
 133 PRINT"YR PRICE COST TOTAL
 135 FORX=1TO10:F=(10000/M)*G:T=T+F
 140 G=INT(G*100)/100:T=INT(T*100)/100
 150 F=INT(F*100)/100:PRINTX;G;F;T:G=G+G*I:NEXT
ready
```

THE GARBAGE COLLECTOR BUG

(THE PROBLEM AND A SOLUTION)

IT IS INTERESTING, BUT INFURIATING, THAT A SERIOUS BUG STILL EXISTS UNCORRECTED IN THE VERSION OF MICROSOFT'S 6502 BASIC USED BY OSI - THE 'GARBAGE COLLECTOR' BUG THAT CONVERTS EVERYONE'S WORD PROCESSOR INTO AN UNUSABLE MESS OF GARBAGE. OHIO SCIENTIFIC, WE ARE TOLD, HAS NO REAL PLANS TO FIX THE PROBLEM, SINCE "'THE SMALL BENEFITS' ARE TOTALLY OUT OF PROPORTION TO THE COST OF MASKING NEW ROMS".

LIKE MOST BUGS, IT'S SMALL, SUBTLE, NOT OFTEN ENCOUNTERED, BUT ALMOST INVARIABLY FATAL TO THE PROGRAM CONCERNED. IT ONLY OCCURS WHEN BASIC TRIES TO RESHUFFLE STRING ARRAYS TO REMOVE REDUNDANT ONES AND 'INVENT' A LITTLE MORE ROOM - HENCE THE TERM 'GARBAGE COLLECTION', AND HENCE ITS FATAL EFFECT ON WORD PROCESSOR PROGRAMS. THE SYMPTOMS ARE WELL KNOWN: IN THE MIDDLE OF HANDLING A STRING ARRAY THE PROGRAM SUDDENLY 'HANGS' , AND THE SCREEN SEEMS TO 'PULSE' ABOUT ONCE EVERY ONE AND A HALF SECONDS. SOMETIMES, BUT ONLY SOMETIMES, THE SYSTEM RECOVERS - AND THEN ONLY AFTER A LONG WAIT; BUT EVEN THEN THE CONTENTS OF THE STRING ARRAYS WILL USUALLY HAVE BEEN SCRAMBLED INTO GARBAGE THEMSELVES. THE SAME THING HAPPENS IF THE 'HOW MUCH FREE MEMORY?' FUNCTION (Y=FRE(0)) IS CALLED.

THIS ONLY HAPPENS WITH STRING ARRAYS SUCH AS A$(1), A$(2), NOT WITH SIMPLE STRINGS LIKE A1$ AND A2$; AND IT TENDS MOSTLY TO HAPPEN WHEN STRING ARRAYS ARE CONCATENATED - SUCH AS USING A$(X)=A$(X)+B$ TO BUILD UP A STRING, SINCE THESE OPERATIONS USE UP A VAST AMOUNT OF TEMPORARY STORAGE SPACE WHILE THE STRING IS BUILT. TO DEMONSTRATE WHAT HAPPENS, GEORGE CHKIANTZ PROVIDED US WITH THIS MODIFICATION OF A ROUTINE ORIGINALLY PUBLISHED IN THE FIRST BOOK OF OSI BY WILLIAMS AND DORNER. IT STORES THE STRINGS IN SCREEN MEMORY RATHER THAN IN THE NORMAL PROGRAM WORKSPACE; AND THEN SHOWS WHAT HAPPENS AS A STRING IS BUILT UP.

FIRST, CHANGE THE STRING SPACE POINTERS IN IMMEDIATE-MODE (I.E. TYPE IN THE POKES WITHOUT ANY LINE NUMBERS). THIS CAN'T BE DONE WITHIN A PROGRAM - IT WOULD LOSE ALL ITS VARIABLE AND STRING POINTERS IN THE PROCESS!

POKE123,0:POKE124,209:POKE133,0:POKE134,212 (OR POKE 134,216 FOR A 2K SCREEN MEMORY DISPLAY ON A C2/4/8).

THEN USE A PROGRAM TO CLEAR THE SCREEN, BUILD UP A STRING ARRAY (IN THIS CASE OF THE ALPHABETIC CHARACTERS A TO Z REPEATED FOR EACH ELEMENT IN THE ARRAY), AND HALT BETWEEN BUILDING EACH ELEMENT BY CALLING THE KEYBOARD ROUTINE, TO WAIT UNTIL ANY KEY IS PRESSED.

```
10 FORI=1TO32:?:NEXT:REM SCREEN CLEAR
20 DIML$(20)
30 K=64
40 FORI=1TO26
50 FORJ=1TOK:L$(I)=L$(I)+CHR$(96+J)
```

```
60 REM SEE WHAT HAPPENS IF YOU INSERT Y=FRE(0) HERE!
70 NEXTJ
80 POKE11,0:POKE12,253:X=USR(X):REM WAIT UNTIL ANY KEY IS PRESSED TO CONTINUE.
90 NEXTI
```

WHEN THIS PROGRAM IS RUN, THE SCREEN WILL FILL WITH 'GARBAGE' STRINGS FROM THE CONCANTENATION - THE BOTTOM STRING WILL BE THE FINAL CORRECT ONE. THE PROGRAM WILL THEN WAIT FOR ANY KEY TO BE PRESSED, ON WHICH IT WILL CONSTRUCT ANOTHER STRING AS THE NEXT ELEMENT IN THE ARRAY. THIS WILL CONTINUE UNTIL THE STRING SPACE IS FULL AND THE STRINGS BEING STORED MEET UP WITH THE POINTERS AT THE TOP OF THE SCREEN. IF THE GARBAGE COLLECTOR (OR GC FROM NOW ON) DID ITS JOB PROPERLY, ALL THE 'GARBAGE' WOULD BE CLEARED AS STRING SPACE RAN OUT, AND ONLY REAL STRINGS WOULD REMAIN, EVENTUALLY CAUSING AN 'OM ERROR' (OUT OF MEMORY) WHEN THEY RAN INTO THE POINTERS. IN PRACTICE, WITH THE ORIGINAL GC, THE ROUTINE CAN ONLY FILL THE AREA - IT 'BOMBS OUT' AS SOON AS THE GC IS CALLED, EITHER ON RUNNNING OUT OF STORAGE SPACE, OR IF FRE(0) IS CALLED.

THAT IS WHAT GOES WRONG, AND THAT IS THE PROBLEM WITH TRYING TO WRITE ANY KIND OF WORD-PROCESSOR FOR THE BASIC IN ROM. QUITE SIMPLY, IT DIES. THERE ARE A NUMBER OF 'FIXES' AROUND, SUCH AS THOSE PUBLISHED BY ELCOMP AND AARDVARK IN THEIR RESPECTIVE 'FIRST BOOK OF OHIO SCIENTIFIC' AND 'FIRST BOOK OF OSI'. ELCOMP'S ROUTINE DOES NOT WORK AT ALL, WHILE AARDVARK'S BASIC PATCH ONLY IMPROVES THE SITUATION RATHER THAN RESOLVING IT. THE ONLY COMPLETE SOLUTION IS TO FIX THE PROBLEM AT THE MACHINE CODE LEVEL, OF WHICH MORE ANON; BUT FOR THE MOMENT, HERE IS THE AARDVARK ROUTINE. AS CAN BE SEEN WITH THE TEST PROGRAM ABOVE, IT RUNS A LOT LONGER BEFORE EXPIRING - PROBABLY GOOD ENOUGH FOR MANY APPLICATIONS.

```
 10  X=PEEK(133):Y=PEEK(134)
 20  L=256*Y+X:L=L-262
 30  Y=INT(L/256):X=L-256*Y
 40  POKE 133,X:POKE 134,Y
 50  POKE 11,X:POKE 12,Y
 60  PRINT "POKE 11,";X;"POKE 12,";Y
 70  PRINT L: A=45383:B=45644
 80  K=L:FOR I=A TO B
 90  IF I<>A+34 THEN 110
100  M=K+146:GOTO 240
110  IF I=A+59 THEN 130
120  M=K+141:GOTO 240
130  IF I=A+67 THEN POKE L,4: GOTO 230
140  IF I<>A+84 THEN 160
150  M=K+209:GOTO 240
160  IF I<>A+137 THEN 180
170  M=K+146:GOTO 240
180  IF I=A+216 THEN POKE L,2:GOTO 230
190  IF I=A+217 THEN POKE L,24:GOTO 230
200  IF I<>A+261 THEN 220
210  M=K+4:GOTO 240
220  X=PEEK(I):POKE L,X
230  L=L+1:NEXT:PRINT "Location":END
240  Y=INT(M/256):X=M-256*Y
250  POKE L,Y:POKE L-1,X
260  GOTO 230
```

LETTERS (OH, DO WE GET LETTERS!)

SAMUEL SHIVE, MANDARIN, FL

SOME QUESTIONS: 1) WHAT, SPECIFICALLY, IS THE COMPUTER DEMONSTRATION/GAME CALLED LIFE? EVERYONE SEEMS TO ADVERTISE IT BUT NOBODY SAYS WHAT IT DOES.

2) ACCORDING TO ONE OF YOUR RECENT JOURNALS, OSI BASIC HAS SOME KIND OF STRING BUG WHICH REQUIRES A GARBAGE COLLECTOR ROUTINE. WHAT BUG?

3) HAVE YOU TESTED THE EXATRON STRINGY FLOPPY YET? ANY INTERFACE PROBLEMS?

4) WHAT DO THE ABBREVIATIONS PIA AND ACIA MEAN?

5) DOES THE ADDITION OF YOUR NEW 8K BOARD REQUIRE EXTRA POWER AND/OR COOLING SYSTEMS (LIKE FANS)?

(A FEW ANSWERS: 1) THE GAME OF LIFE IS A LIFE SIMULATION INITIALLY CONCOCTED BY A BIOLOGY PROFESSOR. IN THE PRE-COMPUTER AGE, IT WAS PLAYED ON A CHECKER BOARD WITH CHECKERS. THE RULES ARE SIMPLE, AND DO PROVIDE AN INTERESTING SIMULATION OF LIFE PROCESSES OF COLONY ORIENTED LIFE FORMS. THE GAME IS PLAYED ON A GRID OF ANY SIZE AND INITIATED BY PLACING COUNTERS ("COLONIES") IN ANY DESIRED PATTERN ON THE BOARD. IN EACH TURN ("GENERATION"), NEW COLONIES ARE BORN AND SOME OLD COLONIES DIE ACCORDING TO A SIMPLE SET OF RULES. IF A COLONY IS SURROUNDED BY LESS THAN TWO OTHER COLONIES, IT DIES OF LONELINESS. IF IT HAS FOUR OR MORE NEIGHBORS, IT DIES OF OVERCROWDING. IF ANY SQUARE IS SURROUNDED BY EXACTLY THREE OTHER COLONIES, A NEW COLONY IS BORN IN THAT SQUARE.

FOR SEVERAL YEARS, INCLUDING A FEW OF THE ONES I SPENT IN COLLEGE, A PAPER-BURNING PASTTIME FOR STUDENTS WAS THE INVENTION OF NEW LIFE FORMS. IT WAS DISCOVERED THAT IF YOU WERE TO PLACE

CERTAIN PATTERNS OF COLONIES ON THE BOARD, THEY WOULD MOVE ACROSS THE BOARD, SHOOT OUT TENTACLES, EAT THINGS IN THE WAY, OR FLASH BACK AND FORTH IN A CYCLIC MOTION. MORE RECENTLY, AN INTERESTING INNOVATION HAS BEEN TWO-MAN LIFE WHERE EACH PERSON TRIES TO PLACE COLONIES THAT WILL DESTROY HIS OPPONENT'S COLONIES.

ABOUT TWO YEARS AGO, BYTE MAGAZINE CAME OUT WITH A MULTI-PAGED ARTICLE CONTAINING AN EXECELLENT DESCRIPTION OF THE GAME AND A NUMBER OF INTERESTING STARTER PATTERNS. I MIGHT SUGGEST THAT THE FREE LIFE GAME IN OUR LARGE CATALOG IS AS GOOD AS ANY FOR A STARTER TO SEE HOW THE PROCESS WORKS. IF YOU DECIDE TO GET SERIOUS AND DO IT PROPERLY, A MACHINE CODE SUBROUTINE IS REQUIRED TO SPEED UP THE GAME.

2) THE OSI GARBAGE COLLECTOR BUG SHOWS UP WHEN YOU ARE USING SUBSCRIPTED VARIABLES AND CONCATENATING STRINGS. WHEN THE TEMPORARY STRING STORAGE AREA IS FULL, BASIC CALLS A ROUTINE WHICH IS SUPPOSED TO COMPACT ALL OF THE CURRENTLY VIABLE STRINGS INTO THE TOP OF THE AVAILABLE MEMORY. UNFORTUNATELY, THERE ARE A COUPLE OF BUGS IN THE ROM, AND IT, THEREFORE, GETS LOST AND NEVER COMES BACK AGAIN. IT CAUSES THE SCREEN TO FLASH AND THE KEYBOARD TO LOCK UP.

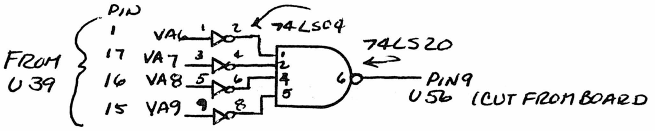
3) NO, I HAVEN'T TESTED THE EXATRON MINI-FLOPPY YET. I WAS APPROACHED BY EXATRON AT ONE COMPUTER SHOW WITH THE SUGGESTION THAT I WOULD WANT TO ADAPT ONE OF THEIR STRING FLOPPYS TO THE C1P. AFTER COMPARING THE PRICE OF THE STRINGY FLOPPY TO A DISK SYSTEM AND LOOKING AT THE AMOUNT OF WORK INVOLVED IN DEVELOPING INTERFACE ROUTINES, IT SEEMED EVIDENT TO ME THAT THE STRINGY FLOPPY WAS A VERY BAD IDEA. IT DOES, AFTER ALL, NOT GIVE REAL FILE CAPABILITIES, PROVIDES NO RANDOM ACCESS, AND COSTS DARN NEAR AS MUCH AS A DISK - WHICH, OF COURSE, WOULD COME WITH AN ALREADY PREPARED OPERATING SYSTEM.

4) THE ABBREVIATION PIA MEANS "PARALLEL INTERFACE ADAPTOR". ACIA IS A FUN ONE - IT MEANS "ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTOR.

5) THE NEW 8K MEMORY BOARD SHOULD NOT REQUIRE EXTRA POWER AND COOLING SYSTEMS. AS OSI POWER SUPPLIES ARE RATED AND HAVE ENOUGH POWER TO DRIVE BOTH THE SUPERBOARD AND THE 8K RAM BOARD, PROVIDED YOU USE LOW POWER CHIPS. HOWEVER, SINCE I DO NOT DEPEND ON COMPUTER SPECIFICATIONS, I WOULD PERSONALLY RECOMMEND THE ADDITION OF A COOLING FAN. ALSO, WE HAVE MADE PROVISIONS ON THE BOARD FOR HOOKING IT TO AN EXTERNAL POWER SUPPLY SHOULD YOUR C1P BE UNABLE TO HANDLE THE LOAD.

AL MC CANN JR., 30 S. ELMWOOD AVE, GLENOLDEN, PA 19036

HERE IS A SMALL HARDWARE FIX I DEVELOPED TO ELIMINATE THE ANNOYING DIAGONAL DOTS ON A C1P VIDEO DISPLAY.



TO CONNECT TO PIN 9, U56, IT MUST BE SEPARATED FROM THE BOARD. I CUT IT (CAREFULLY!!) AT THE BORD WITH A VERY SMALL PAIR OF WIRE CUTTERS. BEND IT HORIZONAL TO SOLDER THE WIRE TO IT. IF YOU ARE USING A CHEAP MONITOR (LIKE ME), OR AR USING R.F. ENTRY, YOUR TV HAS OVERSCAN. THAT MEANS THAT THE DISPLAY FALLS OFF THE SCREEN AT THE TOP AND BOTH SIDES. WHEN IT FALLS OFF THE TOP, IT WINDS UP THE RETRACE LINES. THE CIRCUIT SHOWN BLANKS THE FIRST TWO FULL 32 CHARACTER LINES. D000 - D03F ARE THE ADDRESSES THAT ARE BLANKED.

LOUIS BEER, PORTOLA, CA

HERE'S A TIP THAT MIGHT HELP OTHER PROGRAMERS INTERESTED IN CRYPTANALYSIS. IT IS OFTEN NECESSARY TO KNOW WHETHER A SUBSCRIPT IS ODD OR EVEN. I DO IT THIS WAY:

```
100 IFINT(X/2)<X/2THEN......:REM X IS ODD.
```

JOSEPH ENNIS, NICEVILLE, FL

BESIDES USING 'WAIT' AS AN 'AND' FUNCTION TO RESUME WHEN A SPECIFIC EVENT HAPPENS (EXAMPLE: WAIT61441,1 DETECTS INPUT TO THE CASSETTE INTERFACE), WAIT CAN BE USED AS A 'NAND' FUNCTION WHICH WILL RESUME WHEN ANYTHING ELSE BUT A CERTAIN THING HAPPENS. FOR EXAMPLE, WAIT 57100,254,254 (SUPERBOARD, C1 VALUES) WILL ONLY RESUME WHEN ANY OF THE FOLLOWING KEYS ARE PRESSED IN THE KEYBOARD: ESC, CTL, L-SHIFT, R-SHIFT, AND REPT. THIS BEATS HAVING TO WRITE A POLLING LOOP. A PEEK ON THE SAME LINE WILL RETRIEVE THE ASC VALUE OF THE KEY PUSHED. WAIT 57100,254,254:P=PEEK(57100) WILL STORE THE VALUE OF P AS IN THE FOLLOWING TABLE (C1, SUPERBOARD VALUES):

ESC	222
CTL	225
L-SHIFT	250
R-SHIFT	252
RPT	126

(C2 VALUES = 255 - C1 VALUES)

A POSSIBLE USE IS A HANDY ONE KEY JUMP TO EDIT SUBROUTINES. THE LINE FOLLOWING THE WAIT CAN BE A COMBINATION OF CONTINUE OR GOSUB ON AN IF...THEN TEST ON AN ARITHMETIC 'AND' STATEMENT, OR A COMPUTED GOTO STATEMENT, SUCH AS 'ON1+(LOG(ABS(254-P))/LOG(2))GOTO... THIS WILL RETURN FIVE INTERGERS WHICH ARE 1,2,3,6,AND 8, WHICH MEANS THAT JUMPS 4 AND 7 MUST BE DUMMIED. WAIT57088,254,254 DOESN'T WORK AS WELL. IT RESUMES O.K. BUT SO FAR I NEED LOTS OF LINES TO PUT THE PROPER VALUE IN P.

SCOTT KLAVON, MILWAUKEE, WI

RECENTLY I PURCHASED A VIDEO MOD FOR MY C1 FROM A CANADIAN SOFTWARE HOUSE. THE CATALOG SAID $70.00 FOR 32 X 32 FULL SCREEN INSTEAD OF 24 X 24, EXPANDABLE TO 64 X 64 WITH A FEW EXTRA CHIPS, SWITCH SELECTABLE BAUD RATES AND CPU CLOCK PLUS A NEW CRT DRIVER PROGRAM.

NOW, LET ME TELL YOU THAT I AN AN ELECTRONICS TECHNICIAN AND THAT I KNOW MY STUFF. AFTER A FEW WEEKS, THE KIT ARRIVED AND TWELVE DAYS LATER AFTER MANY HOURS OF FRUSTRATION AND LONG DISTANCE CALLS, ALL AT MY EXPENSE, I REMOVED THE MASS OF JUMPERS AND PIGGYBACKED CHIPS. IT HAD NEVER WORKED, OR EVEN SHOWN ANY SIGN OF WANTING TO. ADMITTEDLY, I HAD MADE A FEW ERRORS, AND THERE WERE SOME IN THE INSTRUCTIONS, BUT EVEN AFTER ALL THE CORRECTIONS AND CHIP REPLACEMENTS (JUST TO MAKE SURE), ALL I COULD EVER GET WAS A BLANK SCREEN. SO AFTER $90 AND A FEW WEEKS OF TIME, I PUT MY MACHINE BACK TO NORMAL.

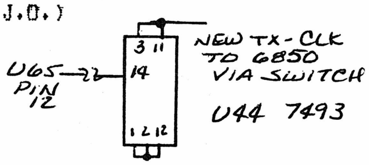
HERE'S SOME ADVICE: 1) DON'T MESS WITH MODS UNLESS YOU ARE CAPABLE OF SUCCESSFULLY INSTALLING AARDVARK'S CT MOD AND LIGHT PEN. THEY MAKE A GOOD STANDARD TO COMPARE BY. 2) IF IT EVER SAYS TO PIGGYBACK A CHIP, DON'T BOTHER GETTING SERIOUS WITH THE MOD, THE DESIGNER DIDN'T. 3) REMEMBER THAT IF YOU EVER DO A MOD AND SOMETHING ELSE GOES HAYWIRE SIX MONTHS LATER, THAT OSI CAN REFUSE TO REPAIR IT.

IT SHOULD BE NOTED THAT I PURCHASED AN INTELLIGENT TERMINAL PROGRAM FOR USE WITH A MODEM FROM THE SAME SOFTWARE HOUSE. IT RUNS, BUT IN SUCH A POOR MANNER THAT IT IS USELESS TO ME. IT SENDS WHAT I TYPE, ALL RIGHT!! FIVE OR SIX TIMES!!!!!

GEORGE MILLER, ROYAL OAK, MI

WE BLEW IT. THE TIP ON HOW TO GET 110 BAUD ON A C1 HAS SEVERAL ERRORS IN THE SCHEMATIC (JOURNAL #4). .

1) PIN 65 SHOULD BE U65
2) THE SCHEMATIC SHOULD SHOW PINS 1, 2, AND 12 TIED TOGETHER, NOT 1, 2, AND 3.
3) THOSE PINS SHOULD JUST BE TIED TOGETHER, NOT RUN TO GROUND. (THAT'S THE LAST TIME I'M DRAWING SCHEMATICS AT 4 AM! WE WERE ABLE TO SET MOST CALLERS STRAIGHT ABOUT 'PIN 65' - THO' MOST ALREADY KNEW WHAT WE MEANT, AND USING PIN 12 INSTEAD OF PIN 3 (MAYBE I ADDED THE 1 AND THE 2 OF TWELVE?????), BUT OUR ORIGINAL HERE SHOWED THE CONNECTION TO GROUND. MAYBE THIS WILL HELP SOME VERY PREPLEXED AND CONFUSED PEOPLE GET THE MOD WORKING. - J.O.)



ROY MILANO, YONKERS, NY

AFTER READING THIS MONTHS ARTICLE ON MEMORY SAVING TECHNIQUES, I MUST SAY THAT THE ARTICLE COVERED ALL THE POINTS WHICH ALLOWS THE PROGRAMMER TO USE ALL THE AVAILABLE RAM THAT HE HAS. I HAVE ALSO FOUND ANOTHER WAY TO SAVE SPACE, ACTUALLY THIS IS A WAY TO INCREASE THE AMOUNT OF RAM AVAILABLE TO THE USER FOR BASIC PROGRAMS. A WORD OF WARNING FIRST, IF YOU INTEND TO USE USR ROUTINES, THEN DON'T USE THIS TECHNIQUE IF YOU LOCATE THEM IN PAGE $02.

THESE ARE THE STEPS TO CHANGE THE SYSTEMS POINTERS IN PAGE $00 THAT WILL ALLOW THE USER TO ACCESS THE 90% OF PAGE 2 WHICH IS NORMALLY UNUSED. (THIS I DISCOVERED WHEN WRITING A 2780 IMULATOR FOR MY C2-4P)

1) COLD START BASIC
2) ENTER INTO MONITOR

3) IN ADDRESS $0222 DEPOSIT $00
4) " " $0079 " $23
5) " " $007A " $02
6) HIT BREAK, WARM START BASIC
7) TYPE 'NEW'

THIS WILL NOW RESET ALL THE SYSTEMS AVAILABLE MEMORY POINTERS SO THAT THE UNUSED PORTION OF PAGE $02 IS NOW ACCESSABLE FOR BASIC PROGRAMS. THIS TECHNIQUE IS VERY HELPFUL WHEN YOU ONLY HAVE A 4K MACHINE.

(NOTE: THIS ROUTINE IS AVAILABLE IN OUR FIRST BOOK OF OSI BY JIM WILLIAMS AND GEORGE DORNER.)

W L MILLER, LIVERMORE, CA

MY C1P HAS SUDDENLY TAKEN TO STATING THAT I HAVE ONLY 753 BYTES FREE AND THEN TELLING ME VERY SOON THAT I HAVE RUN OUT OF MEMORY. WHEN I SPECIFY A MEMORY SIZE OF 5000 IT SAYS 4231 FREE, 3000 - 2231 FREE, AND 10000 - 9231 FREE. I ONLY HAVE 8K. ANY SUGGESTIONS?

(IT SOUNDS AS VERY MUCH LIKE YOU HAVE A BAD MEMORY CHIP IN THE SECOND K OF MEMORY. THE BASIC WOULD SCAN THROUGH FROM THE BEGINNING TO FIND THE END OF MEMORY AND ASSUME THAT THE FIRST PLACE THE MEMORY CHECK FAILED THAT IT WAS OUT OF CHIPS. IT WOULD THEN REMPORT MEMORY ONLY UP TO THE BAD CHIP AND WOULD, OF COURSE, STOP LOADING PROGRAMS AND STOP EXECUTION SPEED WHEN IT HAD USED UP ALL THE MEMORY IT FELT IT HAD AVAILABLE. WHEN YOU SPECIFY A MEMORY SIZE, YOU ARE OVERRIDING BASIC'S AUTOMATIC MEMORY CHECK AND ARE THEREFORE FORCING FALSE ANSWERS. THE 4231 BYTES FREE ON A 5000 BYTE SYSTEM WOULD BE ACCURATE AS OSI RESERVES THE FIRST 769 BYTES OF RAM FOR BASIC SCRATCHPAD WORKSPACE. IT IS THEORITICALLY POSSIBLE TO FORCE THE SYSTEM TO USE THE UPPER 4K OF RAM FOR PROGRAM STORAGE BY CHANGING THE POINTERS TO THE LOW END OF BASIC WORKSPACE. I WOULD SUSPECT, HOWEVER, IN THIS CASE THAT IT WOULD BE MORE TROUBLE THAN IT IS WORTH. THE SYSTEMS DIAGONISTIC WHICH IT RUNS ON START UP TO GET THE MEMORY SIZE IS ALREADY PINPOINTING THE BAD CHIP OR BAD SOCKET FOR YOU. I MIGHT POINT OUT THAT A BAD SOCKET ON THAT SECOND K OF RAM IS ALMOST AS LIKELY AS A BAD CHIP. - R.O.)

PETER KELLNER, RIVERDALE, NY

RECENTLY, WHILE USING MY C2 ROM BASE MACHINE, I DISCOVERED AN INTERESTING QUIRK OF OSI ROM BASIC. ON SEVERAL OCCASIONS, WHILE LOADING A TAPE WITH SOME NOISY SECTIONS, THE LISTING ON THE SCREEN WOULD STOP, BUT THE REMAINDER OF THE PROGRAM ON THE TAPE WOULD CONTINUE TO LOAD UNTIL THE END. AT THE END OF THE LOADING, THE STANDARD SYNTAX ERROR MESSAGE AND THE OK PROMPT WOULD SHOW UP ON THE SCREEN.

APPARENTLY, NOISE ON THE TAPE CAUSE A CHARACTER TO CHANGE TO DECIMAL 143, WHICH JUST HAPPENS TO BE THE BASIC TOKEN FOR 'STOP'. ALSO THE SWITCH HAPPENED BETWEEN TWO QUOTES IN A PRINT STATEMENT, SUPPRESSING THE SYSTEM'S URGE TO CONVERT IT BACK TO A REGULAR TOKEN. THIS BUG CAN EASILY BE EXPLOITED TO SUPPRESS A LISTING OF A PROGRAM SO AS TO AVOID GIVING AWAY VITAL CLUES AS IN AN AARDVARK ADVENTURE.

TO IMPLEMENT THIS FEATURE IN A PROGRAM TAPE, SIMPLY PRECEED YOUR PROGRAM WITH A LINE SUCH AS - `10 ?"PROGRAM DOES NOT APPEAR ON SCREEN, WAIT FOR OK PROMPT!!!!!".` USE `'FORI=769TO850:?I,CHR$(PEEK(I)) :NEXT.` THIS WILL GIVE YOU THE LOCATIONS FOR EACH OF THE LETTERS IN LINE 10 GIVE OR TAKE A FEW. PICK OUT THE MEMORY LOCATION OF ONE OF THE EXCLAMATION MARKS (FOR EXAMPLE 828) AND USE A DIRECT COMMAND TO POKE 828, 143. THIS PUTS A STOP TOKEN IN THE LINE. THE PROGRAM MAY BE SAVED AS USUAL, BUT WHEN RELOADING IT WILL LIST AS FAR AS LOCATION 828 AND THEN STOP WHILE THE PROGRAM CONTINUES TO LOAD.

DAVID PITTS, HOUSTON, TX

I FOUND A TYPO IN GOMOKU (ISSUE #3)! LINE 730 SHOULD READ

```
730E=F(N(3)+1):GOTO750
```

(I.E. AN F INSTEAD OF A G)

ALSO, MY C4P CONTINUES TO AMAZE ME WITH ITS SPEED AND EASE OF PROGRAMMING. AT SOME POINT IN THE FUTURE I WOULD LIKE TO IMPROVE THE SCREEN RESOLUTION SO THAT I CAN USE IT FOR HIGH RESOLUTION GRAPHICS, ETC. A POSSIBILITY I HAVE THOUGHT OF IS TO INTERFACE WITH THE S-100 ELECTRIC CRAYON.

RONALD LASHLEY, MIRAMAR, FL

THE FOLLOWING CHANGES TO ALIEN II MAKE THE GAME MORE CHALLENGING. (VALUES FOR C2/4).

FOR JOYSTICK CONTROL:

```
     470 FORZ1=1TOSM:POKEKE,128
     480 K=PEEK(KE)AND31:IFK=4ORK=5THENPOKEGP,BL:GP=GP+1:POKEGP, GU:GOTO510
     500 IFK=2ORK=3THENPOKEGP,32:GP=GP-1:POKEGP,GU
     510 IFNOTSFANDK=1THEN530
```

TO IMPROVE THE ALIEN'S EVASIVE TACTICS AT LEVELS 3, 4, AND 5:

```
     465 IFRND(G)>.8ANDG<4THEN900
```

TO FIX BUG AT END OF GAME (AFTER THREE RACKS)

```
     1260 TU=TU+1:GOTO990
```

WALTER THOMAS, LINDEN, PA

I HAVE A C1P (600 BOARD) WITH 610 BOARD INSTALLED AND 24K OF MEMORY. I HAVE TWO QUESTIONS:

1) HOW DO I USE THE CLOCK ON THE 610 BOARD (REAL TIME)?

2) WHY CAN'T OSI C1P USERS USE THE MPI 51 DISK SYSTEM WHICH IS UNFORMATTED FOR $200 LESS THAN OSI DEALERS WANT FOR THE SAME MACHINE FORMATTED (OR IS IMPOSSIBLE TO FORMAT THE MACHINE)?

(IF YOU HAVE A SAMS PHOTOFAX FOR YOUR SYSTEM, I CAN GIVE YOU GENERAL INFORMATION ON HOW TO USE THE ON-BOARD CLOCK ON THE 610 BOARD. UNFORTUANATELY, A SPECIFIC LIST OF INSTRUCTIONS WOULD BE A LITTLE TOO LENGTHY FOR THIS COLUMN TO HANDLE. IF YOU LOOK AT THE CLOCK SECTION ON THE 610 BOARD, YOU WILL SEE THAT THERE ARE SEVERAL OUTLETS LABELED IN TIME UNITS. THESE EACH OUTPUT A PULSE AT THE TIME INTERVALS MARKED ON THE SCHEMATIC. TO USE THE CLOCK ON THE BOARD, YOU HAVE TO WIRE (THROUGH A SWITCH) ANY ONE OF THOSE CLOCK OUTPUTS TO THE NMI LINE WHICH IS PHYSICALLY CLOSE TO THE CLOCK ON THE BOARD. WHEN THE CLOCK BOARD GOES HIGH, IT WILL FORCE A JUMP TO THE NMI ROUTINE WHICH IS POINTED TO BY THE LAST TWO BYTES IN MEMORY. AT THAT LOCATION, YOU MUST PUT IN A SMALL MACHINE CODE ROUTINE WHICH COUNTS THE INCOMING PULSE AND STORES THE COUNT IN A MEMORY ROUTINE WITH AN RTS WHICH CAUSES A JUMP BACK TO PROGRAM. QUITE HONESTLY, IT IS A FAIRLY COMPLEX CONVERSION AND I WOULD NOT SUGGEST THAT YOU DO IT IF YOUR SKILLS ARE SUCH THAT YOU NEED SPECIFIC INSTRUCTIONS ON WRITING THE MACHINE CODE ROUTINE AND WIRING THE SWITCH.

THE MAIN REASON THAT OSI USERS CANNOT USE CHEAPER MPI DISK SYSTEMS IS THAT THE OSI SYSTEM REQUIRES A SEPERATE BOARD CALLED A DATA SEPARATOR BOARD. IT IS AVAILABLE FROM ELCOMP, AND IF YOU HAVE SUFFICIENT TECHNICAL SKILLS, YOU CAN INTERFACE AND ADD IT TO A C1P. I DON'T SUGGEST THAT YOU TRY IT. THE CLEVELAND CONSUMER COMPUTER CENTER CURRENTLY SELLS DISK DRIVES FOR THE C1P COMPLETE WITH SEPERATOR BOARD AND THE PROPER CONNECTORS COMPLETE FOR $299. I DOUBT THAT YOU COULD BUY A MPI DRIVE AND THE NECESSARY HARDWARE FOR LESS. IN FACT, I JUST ORDERED A SPARE DRIVE FROM THEM. - R.G.)

STEVE DONACHIE, MIAMI, FL

TO OBTAIN RIGHT-JUSTIFICATION WHEN USING THE 'PRINT-AT' ROUTINE USE THIS LINE:

```
FOR I=1TOL:POKEA+I-L,ASC(MID$(A$,I,1)):NEXT:REM L IS THE LEN(A$).
```

'A' SPECIFIES WHERE THE LAST CHARACTER OF A$ WILL END UP. IT REALLY MAKES NUMERIC DISPLAYS NICE. I USED IT IN A 10-KEYPAD ENTRY ROUTINE TO SHIFT THE INPUT TO THE LEFT AFTER EACH KEYPRESS --- JUST LIKE A $10.00 CACULATOR, WOW!!

ALSO, A THOUGHT ON THE FACT THAT SPACES CAN BE INCLUDED AT THE END OF A STRING. FOR THE SECURITY MINDED, YOU CAN USE A PASSWORD THAT ENDS IN BLANKS TO LOCK YOUR PROGRAM. IF ANYONE SHOULD SEE THE PASSWORD ON THE DISPLAY, THE ONLY PART VISABLE WILL BE THE LEADING CHARACTERS. BUT THE COMPUTER WILL NOT RECOGNIZE A MATCH UNLESS THOSE CHARACTERS ARE FOLLOWED BY THE CORRECT NUMBER OF BLANKS -- UP TO 71 OF THEM AFTER A ONE CHARACTER MISLEADER!

JEFFRY KAMENETZ, WINDSOR, CONN

IF YOUR TV OVERSCANS BADLY, TRY THE MOVING YOKE TRICK. WITH THE TV OFF (A GOOD LONG TIME, THOSE CAPS CAN HOLD A JOLT FOR A DAY OR SO - J.G.), REMOVE THE BACK OF THE SET AND CAREFULLY NOTE THE POSITION OF THE PICTURE TUBE YOKE AND ITS SLEEVE. LOOSEN THE YOKE AND REMOVE THE SLEEVE. IF IT HAS SOME METAL BACKING, THEN YOU'RE IN LUCK. RETURN THE SLEEVE TO ITS PROPER LOCATION. TURN ON THE TV AND CAREFULLY (VERY CAREFULLY!) SLIDE THE SLEEVE TO THE FRONT OF THE

SET WHILE SIMULTANEOUSLY SLIDING THE YOKE BACKWARDS. THIS SHOULD CAUSE THE WHOLE SCREEN TO SHRINK. IT WILL BE NECESSARY TO READJUST THE VERTICAL LINEARITY AND THE VERTICAL SIZE.

I HAVE TWO SETS AT HOME. ONE HAD A METAL BACKED PAPER SLEEVE THAT ALLOWED ME TO REDUCE THE SCREEN SIZE WITH THIS PROCEDURE. THE OTHER DID NOT HAVE THE METAL FOIL SLEEVE AND WAS NOT ADAPTABLE. SUCH IS LIFE. (IF THAT DOESN'T WORK, REMOVE THE YOKE, WRAP A PIECE OF REYNOLDS WRAP AROUND THE NECK OF THE PICTURE TUBE AND PUT IT BACK ON. CURES OVERSCAN. TOO MUCH CAUSES PICTURE FOLDOVER. WILL NOT GET MORE THAN 24 CHARACTERS ACROSS, BUT WILL GET ALL OF THEM. ED.)

UPDATES

FROM TODD BAILEY, GREENVILLE, OH

IN ORDER TO MAKE THE HIGH SPEED CASSETTE LOADER RUN ON MACHINES WITH OTHER THAN 8K OF RAM, ADD THE FOLLOWING LINES IN PLACE OF LINES 28000 TO 30050:

```
28010 H=PEEK(134)-2:POKE133,205:POKE134,HB:LN=HB*256+205
28020 FORA=LNTOLN+306:READOP:POKEA,OP:NEXTA
28030 POKE542,226:POKE543,HB+1:POKE544,226:POKE545,HB+1:H2=HB+1
28040 POKELN+306,HB:POKELN+303,H2:POKELN+44,H2:POKELN+121,H2
28050 POKELN+114,H2:POKELN+130,H2:POKELN+147,H2:POKELN+153,H2
28060 POKELN+203,H2:POKELN+103,H2:POKELN+259,HB:POKELN+172,H2
```

UPDATING THE C2/4 CURSOR

TO RELOCATE THE C2/4 CURSOR TO WORK ON OTHER MEMORY SIZES THAN 8K, MAKE THE FOLLOWING CHANGES:

(PICK UP THE VALUES FOR 'A', 'B', 'X' AND 'Y' IN THE LOOK-UP TABLE.

IN LINE 190 CHANGE P2=30 TO P2=(X)

IN LINE 200 CHANGE FORIN=7856TO8192 TO FORIN=(A)TO(B)

IN LINES 240, 250, 350, 360, 410, 480, 520, 590, 660, AND 690:

CHANGE ANY 30 TO (X)

CHANGE ANY 31 TO (Y)

LINE 440 - CHANGE ONLY THE FIRST '30' TO (X)

SYSTEM	A	B	X	Y
4K	3760	4096	14	15
12K	11952	12288	46	47
16K	16048	16384	62	63
20K	20144	20480	78	79
24K	24240	24576	94	95

MAKE THE CHANGES TO THE ABOVE LINES AND SAVE ON A FRESH CASSETTE. RELOAD AND VERIFY CORRECT RUNNING. (DON'T RUN BEFORE SAVING AS THE BASIC PROGRAM WIPES ITSELF!) (ED. BOY, I WISH WE COULD FIND THE ORIGINAL LETTER THAT THIS CAME IN. THE AUTHOR DID A GREAT JOB AND I WANT TO MAKE CERTAIN THAT HE GETS HIS GIFT CERTIFICATE AND PROPER CREDIT. HE IS ENTITLED TO A FREE C2E.)

ABOUT THE PROGRAMS

WE'VE ADDED MORE PROGRAMS THIS MONTH AS A FEW SOULS RIGHTFULLY COMPLAINED THAT THERE WASN'T ENOUGH TO DO IN THE LAST JOURNAL. I THINK YOU'LL ENJOY THE CURRENT CROP. A FEW OF THEM REQUIRE SOME WORD OF EXPLAINATION.

ASTROID KILLER IS A PRETTY FAIR GAME JUST THE WAY IT IS. I THINK RICK WIDER DID A NICE JOB PUTTING IT TOGETHER. IT DOES LACK ONE LITTLE FEATURE - NONE OF THE ASTROIDS MOVE. I MIGHT SUGGEST THAT YOU WOULD WANT TO MODIFY THE PROGRAM BY LIMITING THE NUMBER OF ASTROIDS ON THE SCREEN TO TEN , ASSIGN A MOVEMENT FACTOR TO EACH ASTROID AND PUT IT IN A SUBSCRIPTED VARIABLE. YOU THEN COULD CYCLE THROUGH THE ASTROIDS WITH A FOR..NEXT LOOP AND MAKE EACH ONE OF THEM MOVE EVERY TURN. ONCE AN ASTROID WAS BLOWN UP, OF COURSE, YOU WOULD SIMPLY GIVE IT A NEW LOCATION AND MOVEMENT FACTOR AND PUT IT BACK ON THE SCREEN. HAVE FUN!

```
100 REM ASTRO BY RICK WIDEN
110 BL=0:K=0:POKE56900,0:FORX=1TO32:PRINT:NEXT
120 PRINT"    A S T E R O I D"
130 PRINT:PRINT"     A S S A S S I N":PRINT:PRINT:PRINT
140 PRINT"YOUR CONTROLS ARE...":PRINT
150 PRINT"L.SHIFT - CCW ROTATE"
160 PRINT"CTRL    - CW  ROTATE"
170 PRINT"R.SHIFT - ACCELERATE"
180 PRINT:PRINT"BLAST ASTEROIDS & ALIENS":PRINT"WITHOUT A CRACK-UP"
190 PRINT:PRINT:PRINT"ASTEROIDS=1 PT.":PRINT"ALIENS=50 PTS."
200 PRINT:INPUT"ARE YOU AN EXPERT";A$:IFASC(A$)=89THENEX=1
210 REM GET BEAM CHARS AND DIR VECTORS
220 FORX=0TO3:READB(X):READS(X):NEXT:VB=0
230 IFPEEK(57088)<128THENFORX=0TO3:READS(X):NEXT:VB=1
240 DATA139,-32,189,-31,131,1,190,33,-64,-63,1,65
250 FORX=0TO3:B(X+4)=B(X):S(X+4)=-S(X):NEXT
260 FORX=1TO32:PRINT:NEXT:NA=1000:IFVBTHENNA=2000
270 REM INT VIDPNT,VIDTOP,POKE 20 ASTEROIDS,BL=BEAMLOC,L=BEAMLIM,B=BMCHAR
280 P=53775:PT=P-500:FORX=0TO20:POKEPT+NA*RND(8),226:NEXT:X=0
290 REM 310=KEYSCAN. VB=1 IF THIS IS A 540 BOARD (C2/4/8)
300 FORAL=0TO20
310 K=PEEK(57100):IFVBTHENK=255-K
320 PRINTCHR$(13)"       SCORE "SC;
330 IFK<>254ANDK>249THEN420
340 B=B(X):S=S(X):POKEP,26:L=P+S*5:FORBL=P+STOLSTEPS
350 IFPEEK(BL)=226THENSC=SC+1:FORC=0TO15:POKEBL,42:POKEBL,32:NEXT
360 IFPEEK(BL)=4THENSC=SC+50:FORC=0TO50:POKEBL,43:POKEBL,32:NEXT:AF=0
370 POKEBL,B:NEXT:FORBL=PTOLSTEPS:POKEBL,32:NEXT
380 IFAFTHEN510
390 R=RND(8)*NA:IFR<333THENAF=1:PS=PT+3*R:SD=S:GOTO300
400 POKEPT+R,226:GOTO310
410 REM KEYSCN ROUTINE
420 ON(K-248)/2GOTO430,450,470
430 X=X-1:IFX<0THENX=7
440 GOTO340
450 P=P+S:IFPEEK(P)<>32THEN550
460 POKEP,26:FORZ=0TO10:NEXT:POKEP,32:GOTO310
470 X=X+1:IFX>7THENX=0
480 GOTO340
490 REM 430=L.SHIFT. 450=R.SHIFT. 470=CONTROL
500 REM POKE ALIEN IN AND ERASE HIM
510 IFEXTHENPS=PS+SD:PN=PS+S:POKEPS,32:POKEPN,4:SD=S:NEXTAL:AF=0:GOTO530
520 PS=PS-SD:PN=PS-SD:POKEPS,32:POKEPN,4:NEXTAL:AF=0
530 POKEPN,32:GOTO310
540 REM EXPLODE SHIP
550 POKEP,32:FORRAD=1TO6:FORX=1TO7STEP2
560 POKEP+RAD*S(X),195+RND(8)*7:NEXTX
570 FORX=0TO111:NEXT:FORX=0TO7:POKEP+RA*S(X),32:NEXT:NEXTRA
580 D$="NICE GOING CAPTAIN":D=53703:GOSUB640
590 D$="YOUR SCORE IS"+STR$(SC):D=53767:GOSUB640
600 L=53640:R=53657:FORX=0TO1:FORD=LTOR:GOSUB630:NEXTD
610 FORD=RTOLSTEP-1:GOSUB630:NEXTD:NEXT
620 RESTORE:CLEAR:GOTO200
630 POKED,4:FORY=0TO22:NEXTY:POKED,32:RETURN
640 FORY=1TOLEN(D$):POKED+Y,ASC(MID$(D$,Y,1)):NEXT
650 POKED+Y,32:RETURN
```

THIS IS A TOUGH LITTLE GAME. THE TWO DIRECTION CONTROLS TILT THE ROTOR AND DETERMINE WHICH DIRECTION YOU TURN IN.

FOR INSTANCE -

IF YOU ARE GOING UP AND HIT THE LEFT SHIFT, YOU WILL TILT TO THE LEFT AND BE GOING UP/LEFT. ANOTHER LEFT SHIFT WOULD GET YOU GOING LEFT HORIZONTAL.

THE GRAPHICS AREN'T BAD, BUT IT IS LEARNING THE NEW CONTROL SCHEME THAT MAKES THE GAME.

I DID'NT PROVIDE FOR ANY PENALTIES OR LIMITS ON CRASHES.

YOU MIGHT WANT TO DO THAT AS A MINOR PROJECT.

```
100 REM HELICOPTER PILOT
110 REM COPYRIGHTED 1980 BY RODGER OLSEN
120 BT=1000000:DIMD(12):F=1:POKE530,1:REM FOR DISK POKE2073,96
130 INPUT"DO YOU WANT DIRECTIONS";A$:PRINT:PRINT:IFASC(A$)=89THEN950
140 REM DIFFICULTY DETERMINES PERCENTAGE OF BUILDING BLOCKING PATH
150 INPUT"HOW GOOD ARE YOU (1-9)";DI:DI=DI/40
160 REM BOT=BOTTOM OF SCREEN. OP=TOP OF SCREEN. TI=TIME CONSTANT
170 REM L=LINE LENGTH. W=VISIBLE SCREEN WIDTH. CC,CL,SL,FA AND ST
180 REM ARE KEY VALUES RETURNED FROM 57100-AR=ARROW DISPLAY
190 REM AR(X)=ARROW DISPLAY VALUES TO MATCH DIRECTIONS.
200 OP=53376:BOT=54912:SL=0:FA=33
210 L=64:W=L:CC=3:CL=5:TI=50:BL=32:ST=129
220 REM FIND OUT IF IT IS A C1P - GET VALUES IF IT IS.
230 IFPEEK(57100)<127THEN270
240 L=32:W=24:CC=255-CC:CL=255-CL:ST=255-ST:SL=1:FA=255-FA
250 REM TL=TAIL DISPLAY VALUE. TA=TARGET VALUE
260 BO=53989:OP=53413:V=1:AR=53453
270 FORX=1TO40:PRINT:NEXT:TI=100:TL=89:TA=240
280 REM PUT IN SOME BUILDINGS. RANDOM DISTRIBUTION BASED ON DIFFICULTY
290 F=1:FORX=OPTOBOT:IFRND(8)<DITHENPOKEX,161
300 NEXT:G=OP:FORX=0TO8:READAR(X):NEXT:AR=53445
310 REM READ IN MOVE DATA. V=1 THIS IS A C1P SO READ SOME MORE.
320 FORX=1TO12:READD(X):NEXT:IFVTHENFORX=1TO12:READD(X):NEXT
330 REM FIND A PATH TO THE BOTTOM
340 N=D(RND(8)*12+1):IFG+N>OPTHENG=G+N
350 IFG>BOTTHEN390
360 REMBLANK OUT A PATH AT LEAST TWO SQUARES WIDE TO THE VICTIM
370 POKEG,BL:POKEG+1,BL:GOTO340
380 REM POKE UP FIRST HELI DISPLAY
390 HP=BO+L+1+RND(8)*(W-2):G=HP:POKEHP,188:POKEHP+L,89
400 POKE57088,1:IFVTHENPOKE57088,254
401 BL=96
410 REM CENTER LITTLE MAN ON THE TOP OF THE SCREEN
420 M=53248+(5*L)+W/2:POKEM,240
430 REM POKE UP LANDING PLATFORM
440 POKEHP+2*L,30:DI=0
450 REM MOVE BOTTOM DOWN TO ALLOW HELI AND PLATFORM UNDER BUILDINGS.
460 BO=BO+3*L
470 POKEHP+L,TL:POKEHP,188:POKEHP,219
480 REM POKE UP HELICOPTER AND BLADES
490 POKEHP,188:GOSUB680
500 POKEHP,219:GOSUB680
510 REM JUMP TO SPECIAL DIRECTION SCAN IF STARTING (LEFT RIGHT ONLY)
520 IFFTHENGOSUB920:IFF=0THEN580
530 IFFTHEN470
540 REM READ THE KEYBOARD FOR DIRECTION KEYS
550 IFPEEK(57088)=CCTHENDI=DI-1:IFDI<1THENDI=8
```

```
560 IFPEEK(57088)=CLTHENDI=DI+1:IFDI>8THENDI=1
570 IFPEEK(57088)=STTHENDI=0
580 POKEAR,AR(DI)
590 POKEHP,BL:POKEHP+L,BL
600 IFHP+D(DI)>BOORHP+D(DI)<OPTHEN470
610 HP=HP+D(DI)
620 REM PEEK NEW LOCATION FOR LITTLE MAN AND LANDING PLATFORM
630 IFPEEK(HP)=TAORPEEK(HP+L)=240THENTL=241:GOTO470
640 IF(PEEK(HP+L)=30ORPEEK(HP)=30)ANDTL=241THEN800
650 IFPEEK(HP)<>BLORPEEK(HP+L)<>BLTHEN760
660 GOTO470
670 REM ADD UP TIME AND LOOK FOR SPEED CHANGES.
680 FORY=1TOTIME:NEXT
690 IFPEEK(57100)=SLTHENTI=TI+5
700 IFPEEK(57100)=FATHENTI=TI-5
710 REM ADD TO TOTAL TIME (TT)
720 TT=(TI/10)+TT
730 RETURN
740 REM EXPLOSION ROUTINE - EXPLODE, PUT MAN BACK, SET FLAG (F)
750 REM FOR NEW START, PUT TAIL (TL) BACK TO ORIGINAL
760 FORY=1TO3:FORX=0TO32:POKEHP,X:NEXTX,Y
770 POKEHP,BL:HP=G:DI=0:TL=89:POKEM,240
780 F=1:POKEG+2*L,30:GOTO470
790 REM CONGRATULATIONS MESSAGES
800 FORX=1TO30:PRINTTAB(X)"YOU GOT HIM OUT SAFE!!!!!":NEXT
810 PRINT"   TIME WAS   "TT/100:PRINT:PRINT
820 IFTT/100<BTTHENBT=TT/100
830 PRINT"BEST TIME OF THE DAY WAS "BT
840 INPUT"READY FOR ANOTHER MISSION";A$
850 RESTORE:GOTO150
860 REM DATA. FIRST LINE IS FOR ARROWS, SECOND FOR C2/4/8 MOVES
870 REM AND THE THIRD FOR C1P MOVES
880 DATA26,18,17,16,23,22,21,20,19
890 DATA1,-63,-64,-65,-1,63,64,65,64,63,64,64
900 DATA1,-31,-32,-33,-1,31,32,33,32,33,31,32
910 REM SPECIAL MOVE ROUTINE FOR STARTING LEFT AND RIGHT ONLY
920 IFPEEK(57100)=CCTHENDI=1:F=0
930 IFPEEK(57100)=CLTHENDI=5:F=0
940 RETURN
950 PRINT"YOU ARE PILOTING A HELICOPTER
960 PRINT"YOUR JOB IS TO RESCUE THE MAN AT THE TOP OF THE SCREEN
970 PRINT"YOU MUST FLY THE COPTER OVER HIM AND THEN LAND HIM ON
980 PRINT"THE PLATFORM THAT YOU TOOK OFF FROM
990 INPUT"READY FOR YOUR CONTROLS";A$
1000 PRINT:PRINT"<RTSHIFT> TURNS YOU COUNTER CLOCKWISE
1010 PRINT"<L.SHFT> TURNS YOU CLOCKWISE
1020 PRINT"A SMALL ARROW IN THE UPPER LEFT CORNER SHOWS YOUR DIRECTION
1030 PRINT"<ESC> SPEEDS YOU UP. <CTRL> SLOWS YOU DOWN
1040 PRINT"<RPT>STOPS THE CHOPPER"
1050 PRINT:PRINT:PRINT:INPUT"READY TO START";A$:GOTO150
```

```
10 PRINT:PRINT:PRINT:PRINT"MATH PRACTICE
20 PRINT:PRINT:PRINT:INPUT"WHAT IS YOUR NAME";Z$
30 PRINT:PRINT:INPUT"HOW OLD ARE YOU";X
40 PRINT:C=0:T=0:AA=0:PRINT
50 PRINT"THE COMPUTER WILL GIVE YOU 10 PROBLEMS USING ONE MATH SKILL."
60 PRINT:PRINT"WHICH SKILL WOULD YOU LIKE TO PRACTICE?"
70 PRINT
80 INPUT" (1)ADDITION (2)SUBTRACTION (3)MULTIPLICATION (4)DIVISION";B
90 ONBGOSUB170,[illegible],340,420
100 C=C+1:AA=AA[illegible]IFC=10THEN130
110 IFC=10THEN130
120 GOTO90
130 PRINT"YOU ANSWERED";AA;"CORRECT AND HAD";T;"MISTAKES."
140 INPUT"DO YOU WANT TO DO IT AGAIN";D$
150 IFLEFT$(D$,1)="Y"THEN60
160 PRINT:PRINT"IT WAS FUN "Z$" BYE NOW":END
170 Y=11:IFX>6THENY=100:IFX>12THENY=1000
180 E=INT(Y*RND(Y))
190 F=INT(Y*RND(Y))
200 BB=0
210 PRINTE"+"F"=";:INPUTG:PRINT
220 IFG=E+FTHENGOSUB560 :RETURN
230 GOSUB630 :GOTO210
240 RETURN
250 Y=11:IFX>6THENY=100:IFX>13THENY=1000
260 H=INT(Y*RND(Y))
270 I=INT(Y*RND(Y))
280 IFI>HTHEN270
290 BB=0
300 PRINTH"-"I"=";:INPUTJ:PRINT
310 IFJ=H-ITHENGOSUB560:RETURN
320 GOSUB630 :GOTO300
330 RETURN
340 Y=11:W=6:IFX>7THENY=13:W=13:IFX>12THENY=26:W=51
350 K=INT(Y*RND(1))
360 L=INT(W*RND(1))
370 BB=0
380 PRINTK"*"L"=";:INPUTM
390 IFM=K*LTHENGOSUB560 :RETURN
400 GOSUB630 :GOTO380
410 RETURN
420 Y=9:IFX>6THENY=11:IFX>12THENY=99
430 N=INT(100*RND(100))
440 O=INT(Y*RND(1)+1)
450 IFO=0ORO>NTHENGOTO440
460 IFX>12THENR=INT(((N/O)+.005)*100):GOTO500
470 R=INT(N/O)
480 S=INT(((N/O)-INT(N/O))*O)
490 IF(R*O)+S<>NTHENS=S+1
500 BB=0
510 PRINTN"DIVIDED BY"O;:INPUTP
520 IFX>12THENQ=0:P=INT((P+.005)*100):GOTO540
530 INPUT"WITH A REMAINDER OF";Q
540 IFP=RANDQ=STHENGOSUB560 :RETURN
550 GOSUB630 :GOTO510
560 V=INT(5*RND(1))+1
570 IFV=1THENPRINT"THAT'S RIGHT, ";Z$;"!!!"
580 IFV=2THENPRINT"RIGHT
590 IFV=3THENPRINT"WAY TO GO!!!"
600 IFV=4THENPRINT"YOU GOT IT
610 IFV=5THENPRINTZ$;", YOU'RE A SMART COOKIE!!"
620 PRINT:RETURN
630 U=INT(5*RND(1))+1
640 IFU=1THENPRINT"TOO BAD"
650 IFU=2THENPRINT"TRY AGAIN, ";Z$
660 IFU=3THENPRINT"YOU GOOFED"
670 IFV=4THENPRINT"NOPE
680 IFU=5THENPRINT"SORRY, ";Z$;", THAT'S WRONG"
690 BB=BB+1:T=T+1:IFBB=1THENAA=AA+1
700 IFBB=5THENPRINT"LETS TRY A DIFFERENT PROBLEM":GOTO90
710 PRINT:RETURN
ready
```

```
10 PRINT:PRINT:PRINT:PRINT:PRINT"GLEEP
20 PRINT:PRINT"COPYRIGHT 1978  JANE OLSEN
30 PRINT:GOSUB380:INPUT"DO YOU WANT INSTRUCTIONS";A$
35 IFLEFT$(A$,1)="Y"THENGOSUB410
40 GOSUB300
50 LETA=INT(11*RND(10))
60 LETB=INT(10*RND(10)+1)
70 T=0:PRINT:PRINT
100 PRINT"GUESS WHERE THE GLEEP IS"
110 INPUTX,Y
120 LETT=T+1
130 IFX=AANDY=BTHENPRINT"YOU CAUGHT THE GLEEP IN";T;"TRIES":GOTO270
140 IFT>4THEN250
150 IFX<AANDY=BTHENPRINT"GONORTH"
160 IFX>AANDY=BTHENPRINT"GO SOUTH"
170 IFX=AANDY<BTHENPRINT"GO EAST"
180 IFX=AANDY>BTHENPRINT"GO WEST"
190 IFX<AANDY<BTHENPRINT"GO NORTHEAST"
200 IFX<AANDY>BTHENPRINT"GO NORTHWEST"
210 IFX>AANDY>BTHENPRINT"GO SOUTHWEST"
220 IFX>AANDY<BTHENPRINT"GO SOUTHEAST"
230 GOSUB300
240 GOTO100
250 PRINT"YOU DIDN'T CATCH THE GLEEP. THE TRICKY LITTL FELLOW WAS "
260 PRINT"HIDING AT ";A;"AND";B
270 X=A:Y=B:GOSUB300
280 GOTO50
290 GOTO40
300 PRINT"0  1234567890"
310 PRINT"10 =========="
320 FORP=9TO0STEP-1
330 PRINTP;"=========="
340 NEXTP
350 IFT=0GOTO370
360 POKEGL-(LL*X)+Y,32
370 RETURN
380 IFPEEK(57088)>127THENVB=600
390 LL=64:GL=55042:POKE56900,0
400 IFVB=600THENLL=32:GL=54087:RETURN
410 PRINT"YOU GUESS WHERE THE GLEEP "
420 PRINT"IS BY ENTERING 2 NUMBERS"
430 PRINT"SEPERATED BY A COMMA"
440 PRINT"LIKE THIS - 4,5"
450 PRINT"THE NUMBER UT AND DOWN GOES FIRST"
460 PRINT"YOU GET FIVE TRIES TO FIND THE GLEEP"
470 PRINT"NORTH IS TO THE TOP OF THE SCREEN"
480 PRINT"WEST IS TO YOUR LEFT"
490 PRINT"EAST IS TO YOUR RIGHT"
500 PRINT"AND SOUTH IS TO THE BOTTOM"
510 INPUT"INPUT ANY NUMBER TO START";X:RETURN
ready
NEW___LOAD
```

A MYSTERY PROGRAM FOR C1P OWNERS ONLY
SORRY C2/4 PEOPLE. IT CAME IN TOO LATE FOR CONVERSION
AND WE DIDN'T WANT TO DENY C1P PEOPLE SUCH A CUTE
PROGRAM

```
10 REM  C1P CHRISTMAS CARD
20 REM
30 REM   By Paul Iverson
40 REM
50 FOR X=1 TO 25:PRINT:NEXT X
60 PRINT"   MERRY":PRINT:PRINT" CHRISTMAS"
70 FOR X=1 TO 15:PRINT:NEXT X
80 FOR X=1 TO 27:READ A:POKE A,42:NEXT X
90 FOR X=1 TO 9:READ A:POKE A,201:NEXT X
100 FOR X=1 TO 9:READ A:POKE A,200:NEXT X
110 FOR X=1 TO 9:READ A:POKE A,202:NEXT X
120 FOR X=1 TO 9:READ A:POKE A,199:NEXT X
130 FOR X=1 TO 10:READ A:POKE A,191:NEXT X
140 FOR X=1 TO 2:READ A:POKE A,161:NEXT X
150 READ A:POKE A,193
160 RESTORE
170 FOR X=1 TO 27
180 READ A
190 POKE A,96
200 FOR B=1 TO 50:NEXT B
210 POKE A,42
220 NEXT X
230 RESTORE
240 GOTO170
250 DATA 53522,53909,53808,54030,53716,53617,53969
260 DATA 53810,53587,53935,53780,54038,53905,53682
270 DATA 54004,53747,54032,53651,53839,53585,53974
280 DATA 53744,53939,53844,54002,53875,53426
290 DATA 53489,53584,53616,53711,53743,53838,53870.53965,53937
300 DATA 53491,53588,53620,53717,53749,53846,53878,53975,54007
310 DATA 53521,53553,53648,53680,53775,53807,53902,53934,54029
320 DATA 53523,53555,53652,53684,53781,53813,53910,53942,54039
330 DATA 54061,54062,54063,54064.54065,54067,54068,54069,54070,54071
340 DATA 54066.54098,53458

ready
```

** WHAT'S NEW AT AARDVARK **

A LOT!!

** MINIPROS GREW UP TO MAXIPROS. IT'S NOW A FULL GROWN WORD PROCESSOR THAT HAS GLOBAL EDITING, IMBEDDED COMMANDS FOR SINGLE, DOUBLE AND TRIPLE SPACE, FILE ACCESS CAPABILITIES (IT CAN LOOK IN A FILE FOR A NAME AND SIMILAR INFORMATION.), COLUMNAR COMMANDS, AND THE ABILITY TO LINK FILES TOGETHER. IT PRINTED THIS JOURNAL IN ONE PASS. WE ADDED A BUNCH MORE STUFF TOO, LIKE AUTOMATIC PAGING, IMBEDDED COMMANDS FOR MARGIN CHANGES IN THE MIDDLE OF A PRINT AND MORE THAN I CAN LIST HERE.

BEST IF ALL, SINCE IT IS STILL IN BASIC, IT SELLS FOR $39.95

** MINOS IS A NEW AMAZING GAME WITH TREMENDOUS GRAPHICS. IT SHOWS YOU THE INSIDE OF A MAZE WHILE YOU TRY TO FIND YOUR WAY THROUGH IT. CONVINCING ENOUGH TO CAUSE CLAUSTROPHOBIA. BEST GRAPHICS EVER ON AN OSI - MINOS $12.95

** KNAPSACK. PUBLIC KEY CRYPTOGRAPHY. THE BEST CODE SYSTEM EVER INVENTED. TOTALLY UNBREAKABLE. IT CAN, FOR INSTANCE ENCODE MESSAGES SENT BY ELECTONIC MAIL (I.E. SOURCE OR NET> IN UNBREAKABLE FORMAT. AVAILABLE ONLY FOR THE C1P. $15.95

** HOW ABOUT 2 MORE MACHINE CODE LIFE PROGRAMS FOR THE C1P. THE BEST YET. ONE PLAYER VERSION IS $12.95 THE DELUXE FOUR PLAYER GOODY IS $15.95. I WOULD PERSONALLY RECOMMEND THE BETTER ONE.

** A MACHINE CODE SCREEN EDITOR FOR THE C4PMF THAT IS FANTASTIC. EDITS LINES INSTANTLY ANYWHERE ON THE SCREEN. I CANNOT DESCRIBE THIS WELL ENOUGH IN WORDS. $15.95

THERE IS A LOT MORE - SUCH AS NEW GAMES AND MACHINE CODE RENUMBERERS FOR TAPE AND DISK MACHINES. WE ALSO CARRY A LOT OF HARDWARE NOW.

AARDVARK
TECHNICAL SERVICES
1690 Bolton, Walled Lake, MI 48088

BULK RATE
U. S. POSTAGE
PAID
WALLED LAKE, MI
48088
Permit No. 5